Metascalable Quantum Molecular Dynamics Simulations of Hydrogen-on-Demand

Ken-ichi Nomura,[a] Rajiv K. Kalia,[a] Aiichiro Nakano,[a] Priya Vashishta,[a] Kohei Shimamura,[b] Fuyuki Shimojo,[b] Manaschai Kunaseth,[c] Paul C. Messina,[d] and Nichols A. Romero[d]
[a] Collaboratory for Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, University of Southern California, Los Angeles, CA 90089-0242, USA
[b] Department of Physics, Kumamoto University, Kumamoto 860-8555, Japan
[c] National Nanotechnology Center (NANOTEC), Thailand Science Park, Pathumthani 12120, Thailand
[d] Argonne Leadership Computing Facility, Argonne National Laboratory, Argonne, IL 60439, USA
(knomura, rkalia, anakano, priyav)@usc.edu, (143d9003, shimojo)@kumamoto-u.ac.jp, manaschai@nanotec.or.th, (messina, naromero)@alcf.anl.gov

ABSTRACT
We enabled an unprecedented scale of quantum molecular dynamics simulations through algorithmic innovations. A new lean divide-and-conquer density functional theory algorithm significantly reduces the prefactor of the $O(N)$ computational cost based on complexity and error analyses. A globally scalable and locally fast solver hybridizes a global real-space multigrid with local plane-wave bases. The resulting weak-scaling parallel efficiency was 0.984 on 786,432 IBM Blue Gene/Q cores for a 50.3 million-atom (39.8 trillion degrees-of-freedom) system. The time-to-solution was 60-times less than the previous state-of-the-art, owing to enhanced strong scaling by hierarchical band-space-domain decomposition and high floating-point performance (50.5% of the peak). Production simulation involving 16,661 atoms for 21,140 time steps (or 129,208 self-consistent-field iterations) revealed a novel nanostructural design for on-demand hydrogen production from water, advancing renewable energy technologies. This metascalable (or "design once, scale on new architectures") algorithm is used for broader applications within a recently proposed divide-conquer-recombine paradigm.

Categories and Subject Descriptors
J.2 [**Computer Applications**]: Physical Sciences and Engineering – *physics, chemistry.*

General Terms
Algorithms, Performance.

Keywords
Divide-and-conquer, Density functional theory, On-demand hydrogen production.

1. INTRODUCTION: IMPORTANCE OF THE PROBLEM
In broad areas such as physics, chemistry, biology, and materials science, there is urgent need for performing large quantum molecular dynamics (QMD) simulations, which follow the trajectories of all atoms while computing interatomic forces quantum mechanically from first principles [1, 2]. This year marks the 50th anniversary of two seminal developments that underpin the QMD method: The first molecular dynamics (MD) simulation using empirical interatomic forces by Aneesur Rahman [3]; and introduction of the density functional theory (DFT) [4, 5] — the most widely used quantum mechanical (QM) method [6] — for which Walter Kohn received a Nobel chemistry prize in 1998 [7]. By solving N one-electron problems self-consistently instead of directly solving one N-electron problem, DFT approximately reduces the exponential complexity of the quantum N-body problem to $O(N^3)$. DFT-based QMD simulations are typically limited to small systems involving a few hundred atoms due to the asymptotic $O(N^3)$ computational complexity. In special cases where only a small subset of atoms is actively involved in chemical reactions, the computational cost can be reduced by embedding a small QM calculation within a large MD simulation. For the development of such multiscale QM/MM (molecular mechanical) methods [8, 9], Karplus, Levitt, and Warshel shared the 2013 Nobel chemistry prize [10, 11]. Unfortunately, a large number of atoms need to be treated quantum mechanically instead in many important applications.

To overcome the $O(N^3)$ bottleneck in these genuinely quantum-mechanical problems, various $O(N)$ DFT algorithms [12-14] have been designed on the basis of the data locality principle called quantum nearsightedness [15-17]. Among them, the divide-and-conquer (DC) DFT algorithm [18-24] pioneered by Weitao Yang [18] is highly scalable [25, 26] on massively parallel computers. The DC-DFT algorithm represents the three-dimensional space as a union of spatially localized domains, and global physical properties are computed as linear combinations of local domain properties. In the past several years, the DC-DFT algorithm — especially with large basis sets ($> 10^4$ unknowns per electron, which is necessary for the transferability of accuracy [27]) — has at last attained controlled error bounds, robust convergence properties, and adequate energy conservation for its use in QMD simulations, thereby making large DC-DFT-based QMD simulations practical [23, 28, 29].

A major remaining problem associated with DC-DFT is the large prefactor of its $O(N)$ computational cost, which makes it a challenge to perform large QMD simulations, *e.g.*, involving over 10^4 atoms for more than 10^4 time steps. This large prefactor arises from a thick buffer layer that surrounds each computational domain in order to minimize the effect of artificial boundary conditions imposed at domain peripheries [16]. In this paper, we present a new light overhead $O(N)$ DFT algorithm called lean divide-and-conquer (LDC) DFT. On the basis of complexity and error analyses of DC-DFT, LDC minimizes the $O(N)$ prefactor

978-1-4799-5500-8/14 $31.00 © 2014 IEEE
DOI 10.1109/SC.2014.59

through: optimization of DC computational parameters; and a density-adaptive boundary condition. We employ a globally scalable and locally fast (GSLF) electronic-structure solver that hybridizes a global real-space multigrid with local plane-wave bases. Hybrid band-space-domain (BSD) decomposition is used to implement the algorithm on massively parallel computers.

As a result, the LDC-DFT code has achieved a weak-scaling parallel efficiency of 0.984 on 786,432 IBM Blue Gene/Q cores for a 50,331,648-atom (or 39.8 trillion degrees-of-freedom to represent 201,326,592 electrons) system, as well as good strong scaling, 60-times improvement of the time-to-solution over the previous state of the art, and high floating-point performance (50.5% of the theoretical peak, or 5.08 PFLOP/s) on the Blue Gene/Q. Portable performance has been demonstrated on other architectures as well. The LDC-DFT code has been used in 16,661-atom QMD simulation to reveal a novel nanostructural design for on-demand production of hydrogen gas from water using LiAl alloy particles [30]. This finding opens a new avenue toward scalable, rapid and high-yield production of hydrogen gas for renewal energy technologies. As will be elaborated in the conclusion, the global-local separation through LDC and GSLF can be generalized into broad applications within a recently proposed divide-conquer-recombine (DCR) algorithmic paradigm [29]. This will likely make them metascalable [26, 31], assuming only that a tree network topology (involving progressively reduced communication volume at upper tree levels) will be supported.

2. IMPROVEMENT OVER THE PREVIOUS STATE-OF-THE-ART

Electronic structure calculation based on DFT is one of the most extensively studied supercomputing applications. An early precursor includes $O(N)$ semi-empirical QM calculation based on the tight-binding method, which is more approximate than DFT (1994 Gordon Bell prize) [32]. This was followed by $O(N)$ DFT calculation based on the locally self-consistent multiple scattering method (1998 Gordon Bell prize) [33] and $O(N)$ DFT-based QMD simulation (SC01 best paper) [14]. More recent works include four Gordon Bell prizes in the past 8 years [34-37]. These include $O(N^3)$ DFT methods based on plane-wave [34] and real-space grid [37] bases implemented on massively parallel computing platforms such as Blue Gene/L and the K computer. In addition, a novel $O(N)$ DFT algorithm [35] based on a three-dimensional fragment method [38-40] has significantly sped up conventional $O(N^3)$ DFT methods.

In order to quantify the time-to-solution of state-of-the-art DFT calculations, let us consider a product of the number of simulated atoms and the number of self-consistent field (SCF) iterations executed per second. For their 2011 Gordon Bell prize, Hasegawa *et al*. reported an execution time of 5,456 seconds per SCF iteration for a 107,292-atom Si system on the K computer using their $O(N^3)$ real-space grid DFT algorithm [37]. This amounts to 19.7 atom•iteration/s. Very recently, Osei-Kuffuor and Fattebert reported an $O(N)$ DFT calculation, where 1 QMD step took ~275 seconds for a 101,952-atom polymer system on 23,328 Blue Gene/Q cores [41]. Assuming 5 SCF iterations per MD step, this amounts to a speed of 1,850 atom•iteration/s. As will be presented in section 5.2, one SCF iteration for a 50.3 million-atom silicon carbide (SiC) system using our LDC-DFT code took 441 seconds on 786,432 Blue Gene/Q cores, amounting to a speed of 114,000 atom•iteration/s. This represents 5,800- and 62-fold improvements of the time-to-solution over the two previous state-of-the-art calculations mentioned above. Here, care must be taken in these comparisons, since they are between different DFT algorithms (*e.g*. $O(N^3)$ *vs*. $O(N)$) on different platforms. At any rate, the DFT-based QMD simulation presented in this paper is by far the fastest to date. As will be explained in section 3, this progress has been realized by combining the new LDC-DFT algorithm with a number of state-of-the-art computational techniques [34-37]. The improved time-to-solution has allowed us to perform an unprecedented scale of production QMD simulation involving 16,661 atoms for 21,140 time steps (or 129,208 SCF iterations) on 786,432 Blue Gene/Q cores. Most of the largest DFT calculations reported previously were static (*i.e*. one-step) calculations, and we are not aware of any QMD simulation for such long time.

A limited number of parallel $O(N)$ DFT software are available, including CONQUEST [42], ONETEP [43, 44], and SIESTA [45]. Publicly available DC-DFT codes include OpenMX [21], and implementation of DC-DFT on massively parallel computers is discussed in Refs. [26] and [46]. In terms of the parallel efficiency, however, no reported result is higher than that of LDC-DFT in this paper, *i.e*., a weak-scaling parallel efficiency of 0.984 on 786,432 IBM Blue Gene/Q cores. Also, LDC-DFT exhibits a reasonable strong-scaling parallel efficiency of 0.803 on 786,432 Blue Gene/Q cores. Our floating-point performance (50.5% of the theoretical peak on the entire Blue Gene/Q) is within the range of previous state-of-the-art DFT calculations: 56.5% on Blue Gene/L [34], 39% on Cray XT4 [35], 75.8% on Cray XT5 [36], and 43.6% on K [37].

3. APPLICATION AND ALGORITHMIC INNOVATIONS

The performance improvements mentioned above have been achieved through several algorithmic innovations. This section first provides a brief summary of the conventional DC-DFT algorithm [18-24], followed by a description of our algorithmic and computational innovations.

In DC-DFT, the three-dimensional space Ω is represented as a union of overlapping spatial domains, $\Omega = \cup_\alpha \Omega_\alpha$, and physical properties are computed as linear combinations of domain properties (Fig. 1). Each domain Ω_α is further decomposed into its sub-volumes, $\Omega_\alpha = \Omega_{0\alpha} \cup \Gamma_\alpha$. Here, $\Omega_{0\alpha}$ are non-overlapping cores covering Ω (*i.e*., $\Omega = \cup_\alpha \Omega_{0\alpha}$ and $\Omega_{0\alpha} \cap \Omega_{0\beta} = 0$ for $\alpha \neq \beta$), whereas Γ_α is a buffer layer that surrounds $\Omega_{0\alpha}$.

We solve the Kohn-Sham (KS) equations [5] within each domain to obtain local KS wave functions (Eq. (a) in Fig. 2) iteratively using a preconditioned conjugate-gradient (CG) method [47, 48]. Here, $\psi_n^\alpha(\mathbf{r})$ is the n-th KS wave function in domain α with the corresponding energy eigenvalue ε_n^α, ∇^2 is the Laplacian operator, and $\hat{V}_{\text{ion}}$ and $\hat{V}_{\text{Hxc}}$ are the electron-ion and electron-electron (or Hartree-exchange-correlation) potential operators, respectively. Note that the KS Hamiltonian operator $\hat{H}$ in Eq. (a) in Fig. 2, and hence the KS wave functions, depend on the valence electron number density $\rho(\mathbf{r})$, which is a global quantity of the entire system Ω. $\hat{V}_{\text{Hxc}}$ is obtained from $\rho(\mathbf{r})$ in $O(N)$ time using a tree-based algorithm. For each domain Ω_α, we define a domain support function $p_\alpha(\mathbf{r})$ from $\mathbf{r} \in \Re^3$ ($\Re$ is the set of real numbers) to the unit interval [0, 1], which is compactly supported

within the domain, *i.e.*, $p_\alpha(\mathbf{r}) = 0$ ($\mathbf{r} \notin \Omega_\alpha$). The domain support functions constitute a partition of unity, *i.e.*, they satisfy the sum rule, $\Sigma_\alpha p_\alpha(\mathbf{r}) = 1$, at every spatial position $\mathbf{r}$. This allows the global density $\rho(\mathbf{r})$ to be decomposed into partial contributions $\rho_\alpha(\mathbf{r})$ from domains α; see Eq. (b) in Fig. 2. The domain density $\rho_\alpha(\mathbf{r})$ in turn is obtained from the domain KS wave functions $\psi_n^\alpha(\mathbf{r})$ as shown in Eq. (c) in Fig. 2 (where the step function $\Theta(x)$ is 1 and 0 for positive and negative x, respectively). In Eq. (c) in Fig. 2, the chemical potential μ is determined from the total number of valence electrons N in the entire system through the relation, $N = \int d\mathbf{r}\rho(\mathbf{r})$, using the Newton-Raphson method. The global density and local Kohn-Sham wave functions are determined iteratively in an SCF iteration (Fig. 2). Note that the local KS solutions are globally informed through the global density and chemical potential.

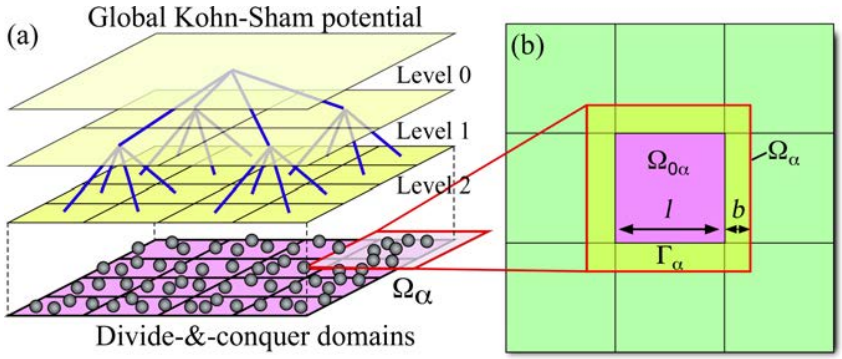

Figure 1. (a) Schematic of divide-and-conquer (DC) density functional theory (DFT). (b) DC domain and buffer.

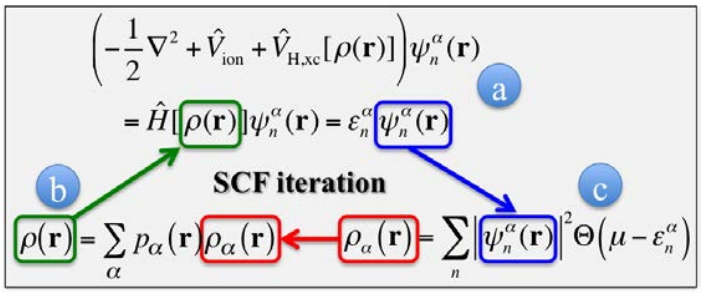

Figure 2. Global-local self-consistent field (SCF) iteration in DC-DFT.

The rest of this section summarizes our algorithmic and computational innovations. Section 3.1 describes the lean divide-and-conquer algorithm to significantly reduce the $O(N)$ prefactor. Our globally scalable and locally fast electronic stricture solver is presented in section 3.2, and we discuss parallelization based on hybrid band-space-domain decomposition in section 3.3. Section 3.4 deals with the optimization of floating-point performance based on algebraic transformation of computations.

3.1 Lean Divide-and-Conquer (LDC)

In order to reduce the prefactor of the $O(N)$ computational cost of DC-DFT, we first optimize the size of the DC domains based on an analysis of its computational cost [29]. Consider a cubic system of side length L in three dimensions, and let the core length and the buffer thickness of a cubic domain be l and b, respectively; see Fig. 1. The computational complexity of the DFT computation within each domain is assumed to be the ν-th power of the system size. The total computational cost is a product of the number of domains and per-domain computational cost, $T_{\text{comp}}(l) = (L/l)^3(l+2b)^{3\nu}$. Given the total system size L and the buffer thickness b, the optimal domain size l_* is determined to minimize $T_{\text{comp}}(l)$ as a function of l, *i.e.*, $l_* = \text{argmin}\, T_{\text{comp}}(l) = 2b/(\nu-1)$ [29]. The computational complexity of the DFT problem is $O(n^2)$ for typical domain sizes, where the number of electrons per domain is $n < 1{,}000$ [34], and thus $l_* = 2b$. The asymptotic complexity, which has rarely been encountered in practical DFT calculations [34], arises from the orthonormalization of KS wave functions and is $O(n^3)$. In this limit, $l_* = b$. In all our applications in this paper, the domains sizes are such that $l_* = 2b$.

The choice of the buffer thickness b is dictated by accuracy requirement. The quantum nearsightedness principle [15] indicates that the error involved in the DC-DFT algorithm (which is due to the artificial boundary condition imposed at the domain boundary $\partial\Omega_\alpha$) decays exponentially as a function of b [16]. Due to the artificial boundary condition at $\partial\Omega_\alpha$, the domain density $\rho_\alpha(\mathbf{r})$ deviates from the total density $\rho(\mathbf{r})$. Let λ be the exponential decay constant of the density perturbation, $\Delta\rho_\alpha(\mathbf{r}) = \rho_\alpha(\mathbf{r}) - \rho(\mathbf{r})$, away from $\partial\Omega_\alpha$ toward the center of the domain. Suppose that an error tolerance of $\varepsilon\langle\rho_\alpha(\mathbf{r})\rangle$ is imposed on $|\Delta\rho_\alpha(\mathbf{r})|$ at the periphery of $\Omega_{0\alpha}$, where $\langle\rho_\alpha(\mathbf{r})\rangle$ is the average density in Ω_α. To satisfy the error tolerance, the buffer thickness needs to be as large as

$$b = \lambda \ln(\max\{|\Delta\rho_\alpha(\mathbf{r})| \,|\, \mathbf{r} \in \partial\Omega_\alpha\} / \varepsilon\langle\rho_\alpha(\mathbf{r})\rangle). \quad (1)$$

Since the computational complexity of the DC-DFT algorithm scales with the buffer thickness asymptotically as $b^{3\nu} = b^6 \sim b^9$

(for $\nu = 2 \sim 3$) [29], the large b value required for obtaining a sufficient accuracy represents a major computational bottleneck. Key to reducing the prefactor of the $O(N)$ computational cost of DC-DFT is thus to minimize $\Delta\rho_\alpha(\mathbf{r})$ at $\partial\Omega_\alpha$.

Our LDC-DFT algorithm addresses this problem through an improved treatment of domain boundaries. According to the Hohenberg-Kohn theorem [4], the local density $\rho_\alpha(\mathbf{r})$ corresponds to a unique external potential $v(\mathbf{r})$, which is distinct from that corresponding to the global density $\rho(\mathbf{r})$. To reduce the discrepancy $\Delta\rho_\alpha(\mathbf{r})$, we use a linear-response formula for the boundary potential,

$$v_\alpha^{\mathrm{bc}}(\mathbf{r}) = \int d\mathbf{r}' \frac{\partial v(\mathbf{r})}{\partial \rho(\mathbf{r}')}\left(\rho_\alpha(\mathbf{r}') - \rho(\mathbf{r}')\right) \cong \frac{\rho_\alpha(\mathbf{r}) - \rho(\mathbf{r})}{\xi}, \quad (2)$$

where ξ is an adjustable parameter. For the last equality in Eq. (2), we have adopted a local approximation [49], which is justified by the quantum nearsightedness principle [15-17] as formulated by Prodan and Kohn [16]. Namely, the response kernel $\partial v(\mathbf{r})/\partial\rho(\mathbf{r}')$ is short-ranged with respect to $|\mathbf{r}-\mathbf{r}'|$. The same density-template potential was used by Ohba *et al*. [24] to augment the hard-wall boundary condition. Here, we instead use the periodic boundary condition on the local KS wave functions, incorporating the boundary potential in the local KS Hamiltonian,

$$\hat{H}_\alpha = -\frac{1}{2}\nabla^2 + \hat{V}_{\mathrm{ion}} + \hat{V}_{\mathrm{Hxc}}[\rho(\mathbf{r})] + v_\alpha^{\mathrm{bc}}(\mathbf{r}), \quad (3)$$

which replaces $\hat{H}$ in Eq. (a) in Fig. 2. It should be noted that the first equality in Eq. (2) is an exact linear response relation according to the foundational theorem of DFT [4]. In this paper, we take the adjustable parameter ξ in Eq. (2) to be 0.333 in the atomic unit by fitting representative systems in Ref. [24].

3.2 Globally Scalable and Locally Fast (GSLF) Solver

To efficiently solve the global-local SCF problem in Fig. 2, our globally scalable and locally fast (GSLF) electronic-structure solver combines a local plane-wave basis within each DC domain for high numerical efficiency and a global real-space multigrid for scalability on massively parallel computers (Fig. 3):

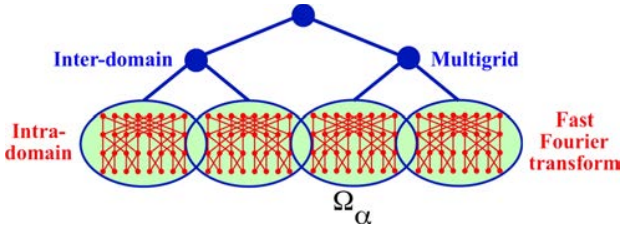

Figure 3. Globally scalable and locally fast, hybrid electronic-structure solver. Blue lines represent a tree network topology for multigrid-based inter-domain computation, whereas red lines show the butterfly network for fast Fourier transform within each domain.

1. ***Fast intra-domain computation***: A plane-wave basis [2, 34, 50] is used to represent local KS wave functions $\psi_n^\alpha(\mathbf{r})$ and charge density $\rho_\alpha(\mathbf{r})$ within each domain (containing ~ 100 atoms), which takes advantage of a highly efficient numerical implementation [51] based on fast Fourier transform (FFT) [52] to solve Eq. (a) in Fig. 2.
2. ***Scalable inter-domain computation***: Once the global density $\rho(\mathbf{r})$ is obtained from $\rho_\alpha(\mathbf{r})$ according to Eq. (b) in Fig. 2, a real-space multigrid method [41] is used to solve the Poisson equation, $\nabla^2 V_{\mathrm{H}}(\mathbf{r}) = -4\pi\rho(\mathbf{r})$, to obtain the Hartree potential $V_{\mathrm{H}}(\mathbf{r})$ [53], which is part of $\hat{V}_{\mathrm{Hxc}}$ in Eq. (a) in Fig. 2. The multigrid method is highly scalable on massively parallel computers due to the locality preserving octree data structure [20, 23]; see Fig. 1(a).

The FFT solver has been used extensively in $O(N^3)$ DFT programs [34, 50]. Plane-wave bases in spatially localized domains are found in $O(N)$ DFT [43, 44] and discontinuous Galerkin [54] methods. The multigrid method has been used in $O(N)$ DFT algorithms that are solely based on real-space representation [20, 23, 41].

3.3 Hierarchical Band-Space-Domain (BSD) Decomposition

An LDC-DFT-based QMD simulation code is implemented on massively parallel computers by employing multiple levels of parallelism with hierarchical band-space-domain (BSD) decomposition; see Fig. 4. (A similar approach was used previously in our space-time-ensemble parallel MD approach [26, 55].) At the coarser level, we use task decomposition among domains Ω_α. The program is implemented using the message passing interface (MPI) library for interprocessor communications, and each domain is assigned a dedicated MPI communicator using an MPI_COMM_SPLIT call. At the finer level, the plane-wave based calculations within each domain are further parallelized by a hybrid approach combining spatial decomposition (*i.e*., distributing real-space or reciprocal-space grid points among processors) and band decomposition (*i.e*., assigning the calculations of different KS wave functions to different processors) [47, 48] within the communicator assigned to the domain. During energy minimization, different KS wave functions are iteratively optimized on different processors independently from the others under an approximate orthonormality condition, *i.e*., band decomposition is employed. The electron density is also calculated by band decomposition. On the other hand, the KS wave functions are orthonormalized by first constructing an overlap matrix between them using

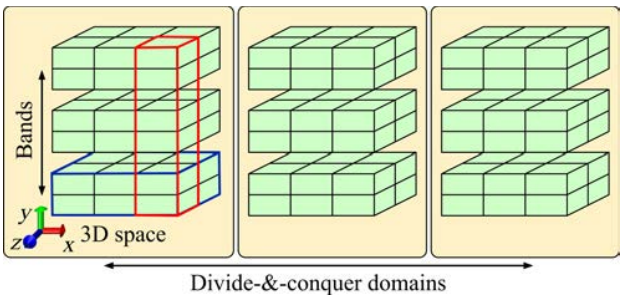

Figure 4. Hybrid band-space-domain decomposition. Within a processor group assigned to each domain, each core computes either a subset of spatial (real or reciprocal) grid points (shown by a red frame) or a band (blue frame) in alternating band-space decompositions. The identical grid *vs*. band decomposition is used in all domains.

reciprocal-space decomposition, where the Fourier components of the wave functions are distributed among multiple processors. This is followed by parallel Cholesky decomposition of the overlap matrix, which introduces an additional parallelization axis. To switch between the spatial and band decompositions, all-to-all communications are required only within the communicator. In addition, global communication within the communicator is necessary to calculate the scalar products between the wave functions.

3.4 Algebraic Transformation of Computations

The numerical kernel of LDC-DFT is the solution of the KS equations (Eq. (a) in Fig. 2). As explained earlier, this is formulated as an iterative minimization of the energy as a functional of KS wave functions using a preconditioned CG method. The original LDC-DFT code used a band-by-band CG minimization, in which KS wave functions (or bands) are optimized one at a time sequentially in ascending order of KS energy. To take maximal advantage of the highly optimized basic linear algebra subprograms (BLAS) library, we instead adopt an all-band approach, in which CG minimization is performed simultaneously for all KS wave functions at once. The original band-by-band computation is expressed in terms of matrix-vector operations using the DGEMV subroutine in level 2 BLAS (BLAS2). By combining all bands together into a matrix and operating the KS Hamiltonian at once, we transform the entire CG minimization to BLAS3 matrix-matrix computations using the DGEMM subroutine. This exposes more parallelism and drastically increases the floating-point performance.

Some of these computational transformations are best carried out in an algebraic manner. Here, we illustrate this approach using one of the most expensive computations in LDC-DFT, *i.e.*, applying the ultrasoft nonlocal pseudopotential operator $\hat{v}_{\mathrm{nl}}$ [47, 48, 56] to the KS wave functions, which is part of the electron-ion potential operator $\hat{V}_{\mathrm{ion}}$ in Eq. (a) in Fig. 2:

$$\hat{v}_{\mathrm{nl}}\left|\psi_n^\alpha\right\rangle = \sum_{I}^{N_{\mathrm{atom}}} \sum_{ij}^{L_{\mathrm{max}}} \left|\beta_{i,I}\right\rangle D_{ij,I} \left\langle\beta_{j,I}\middle|\psi_n^\alpha\right\rangle \quad \left(n = 1,\ldots,N_{\mathrm{band}}\right) \quad . \quad (4)$$

Here, N_{atom} is the number of atoms per domain, L_{max} is the number of angular momenta for projecting KS wave functions onto spherical harmonics centered at the atoms, and N_{band} is the number of KS wave functions. For this computation, each wave function $|\psi_n^\alpha\rangle$ is represented in the reciprocal space as an N_{p}-element vector ($N_{\mathrm{p}} \sim 10^4$ is the number of plane waves). The same representation is used for the angular momentum-dependent projection function $|\beta_{i,I}\rangle$ centered at each atom I. In order to use BLAS3, we rewrite Eq. (4) as

$$\hat{v}_{\mathrm{nl}}\Psi = \sum_{i,j}^{L} \tilde{\mathbf{B}}(i)\tilde{\mathbf{D}}(i,j)\tilde{\mathbf{B}}(j)^{\mathrm{T}}\Psi \quad (5)$$

by packing all of the wave functions into an $N_{\mathrm{p}} \times N_{\mathrm{band}}$ matrix $\Psi = [|\psi_1^\alpha\rangle,\ldots,|\psi_{N\mathrm{band}}^\alpha\rangle]$. In Eq. (5), $\tilde{\mathbf{B}}(i) = [|\beta_{i,1}\rangle,\ldots,|\beta_{i,N\mathrm{atom}}\rangle]$ is an $N_{\mathrm{p}} \times N_{\mathrm{atom}}$ matrix and $[\tilde{\mathbf{D}}(i,j)]_{I,J} = D_{ij,J}\delta_{IJ}$ is an $N_{\mathrm{atom}} \times N_{\mathrm{atom}}$ matrix; $\delta_{IJ} = 1$ ($I = J$) and 0 (else). Consequently, the nonlocal pseudopotential calculation can be performed solely as matrix-matrix operations using the DGEMM subroutine in BLAS3.

4. PERFORMANCE OPTIMIZATIONS

This section first presents the computing platforms, on which the performance of the LDC-DFT code has been tested, followed by the description of thread-level, data-level and I/O optimizations.

4.1 Experimental Platforms

Numerical tests are performed on the IBM Blue Gene/Q computer [57] named Mira at the Argonne Leadership Computing Facility. Mira consists of 48 racks, each with 1,024 nodes. Each node has a 16-core processor operating at 1.6 GHz, 32 MB L2 cache with hardware transactional memory and speculative operation functionalities [58], and 16 GB DDR3 memory. The processor employs the PowerPC A2 architecture that supports quad floating-point processing units (QPUs), 16 KB L1 instruction cache, 16 KB L1 data cache, and 4-way multithreading per core enabling 64 concurrent threads on one node. Though it is highly energy efficient (55 watts per node) due to the relatively low clock speed, the Blue Gene/Q chip delivers a peak performance of 204.8 GFLOP/s. Each node has 11 communication links — 10 links to connect to other computing nodes and one link to an I/O node. Each link can simultaneously transmit and receive data at 2 GB/s, amounting to a total bandwidth of 44 GB/s. A 5-dimensional torus network is used for internode communications [57, 59, 60].

At every clock cycle, each PowerPC A2 core can issue one instruction to the auxiliary execution unit (AXU) for floating-point operations and another to the main execution unit (XU) for load/store, branch and integer operations, *etc*. It thus requires at least 2 instruction streams per core to maximize the instruction throughput. Assigning 4 threads per core could further improve the performance by hiding latency, assuming that the memory bandwidth is not saturated. PowerPC A2 utilizes the quad processing extension (QPX) instruction set for single-instruction multiple-data (SIMD) operations on QPUs, which can perform four double-precision floating-point operations in a single cycle. Therefore, ensuring SIMD utility is critical for achieving decent performance on Blue Gene/Q.

In order to examine performance portability, we benchmark the LDC-DFT code on another architecture, *i.e.*, a computing node comprised of dual Intel Xeon E5-2665 CPUs operating at 2.4 GHz and 64 GB DDR3 memory at the Center for High-Performance Computing and Communication at the University of Southern California. The Xeon E5-2665 employs the Sandy Bridge-EP architecture. There are 8 physical cores and 16 threads supported by the hyper-threading technology. Each core has its own 256 KB L2 cache and shares 20 MB L3 cache per chip with the other cores. Intel turbo boost technology 2.0 and error-correcting-code memory are also supported. Intel QuickPath interconnect provides up to 1,866 mega transfers per second, and the maximum memory bandwidth is 14.9 GB/s per memory channel.

4.2 Architecture-level Optimizations

In addition to the algorithmic and algebraic techniques explained in sections 3, we perform several architecture-level performance optimizations as explained in this subsection. Furthermore, massive file I/O in production runs is handled by collective I/O.

Architecture-level performance optimizations are guided by performance-profiling results. Here, we use the Blue Gene performance monitoring (BGPM) API and the hardware performance monitor (HPM) toolkit for profiling [61]. We have instrumented the LDC-DFT code using three performance counters: (1) PEVT_CYCLES to count CPU cycles; (2)

PEVT_INST_QFPU_FPGRP1_QUAD to count floating-point (FP) operations that utilize the QPUs for SIMD operations; and (3) PEVT_INST_QFPU_FPGRP1_SINGLE for non-QPU FP operations. These three counters are used respectively to identify: (1) cycle hotspots; (2) vectorized code sections; and (3) non-vectorized code sections. The cycle hotspots are marked as candidates for applying thread-level parallelism. On the other hand, code sections with high non-vectorized operation counts are selected for detailed analysis for data parallelism.

Data Parallelism: To ensure that the LDC-DFT code fully utilizes SIMD vectorization in all critical hotspots [62], we have further analyzed the non-vectorized hotspots based on the profiling. The code line-level and function-level profiling results using the HPCTW toolkit [63] have revealed that several performance-critical loop structures (which account for ~20% of the total execution time) have not been vectorized automatically during compilation. Some of these are due to ambiguous data alignment of the complex-variable arrays used in the loop computation. To mitigate this issue, we have manually vectorized five of the most time-consuming non-vectorized loops by creating wrapper subroutines that ensure proper data alignment of complex arrays. In addition, solution of the SCF problem (Fig. 2) based on a plane-wave basis requires efficient FFT operations. On the basis of the profiling, we have found that the commodity FFTW library routines [52] used in the original code consume considerable cycles but mostly from non-SIMD instructions. We have also found that 21.1% of the total cycles, with 72.5% of the total non-vectorized FP operations and less than 1% of the total vectorized FP (QPX) operations, were spent in the FFTW subroutines. This indicates that the computations in the FFTW library on Blue Gene/Q were not SIMD vectorized properly. To alleviate this issue, we have decided to use Spiral, a highly efficient threaded FFT library on Blue Gene/Q [51], instead of the commodity FFTW library used in the original code [52]. Post-optimization profiling results have confirmed that SIMD instructions are now issued in the manually vectorized code sections and in the Spiral FFT calculations.

Thread-Level Parallelism: To fully utilize the hardware threads and double-instruction units on Blue Gene/Q, we have implemented a hybrid MPI+multithreading parallelization scheme [58] in the LDC-DFT code. For multithreading, we use the OpenMP library, where the number of threads is specified by the environment variable, OMP_NUM_THREADS. First, candidate subroutines for threading are selected based on the time spent in each subroutine and its complexity for threading (*e.g.*, severity of race conditions and whether a threaded version of a library is available). This process is assisted by the concurrency control decision tree recently developed by us [58]. On the basis of such prioritization, we apply OpenMP threading to the cycle hotspot loops identified via HPM profiling. We also use the threaded IBM engineering and scientific library (ESSL) for threaded BLAS calculations. In addition, a highly efficient customized threaded matrix-matrix multiplication subroutine (*i.e.* JAG-DGEMM [64]) is used instead of the typical DGEMM in the ESSL package. Finally, FFT calculations are threaded via the Spiral FFT library [51].

Collective File I/O: To address a challenge of massive file I/O on the 786,432-core Blue Gene/Q, we have implemented an aggregated I/O scheme, in which a number of MPI processes are grouped together to perform disk I/O. Within each group, only a master process accesses disk, while the other processes communicate with the master to transfer data. Creating and managing files for each of 786,432 or a larger number of MPI processes is beyond the system's capability, whereas grouping all processes for a single I/O incurs an excessive communication cost and does not take advantage of the multiple I/O servers. We use the optimal I/O group size of 192 MPI processes to minimize the I/O time. As a result of the collective I/O scheme, for a typical production run for about 12 hours on the full 786,432 cores, the file read and write times of 9.1 and 99 seconds account for only 0.02% and 0.23%, respectively, of the total execution time. To reduce the I/O size for atomic coordinates, we can optionally use our spacefilling-curve-based adaptive data compression scheme [65], though the compression ratio is rather small for the 16,611-atom production run shown in section 6.

5. SCALABILITY, TIME-TO-SOLUTION, AND PORTABLE FLOATING-POINT PERFORMANCE

The algorithmic innovations in section 3 and performance optimizations in section 4 have resulted in excellent scalability, time-to-solution, and portable floating-point performance of the parallel LDC-DFT algorithm. The program is written in Fortran 90 with MPI for message passing, combined with OpenMP for multithreading. In the following, we present performance measurement results of the LDC-DFT code. Section 5.1 presents scalability data including both weak- and strong-scaling results. The time-to-solution and peak floating-point performance are presented in sections 5.2 and 5.3, respectively. Section 5.4 deals with performance portability, and this section concludes with the verification and validation of the LDC-DFT code in section 5.5.

5.1 Weak and Strong Scaling

We first perform a weak-scaling benchmark of the LDC-DFT code on the Blue Gene/Q, in which the number of atoms per core N_{atom}/P is kept constant. Figure 5 shows the wall-clock time per QMD simulation step with scaled workloads — 64*P*-atom SiC system on *P* cores of Blue Gene/Q. In this test, we set the number of domains to be *P*. The execution time includes 3 SCF iterations to determine the KS wave functions and the global density, with 3 CG iterations per SCF cycle to refine each wave function. By increasing the number of atoms linearly with the number of cores,

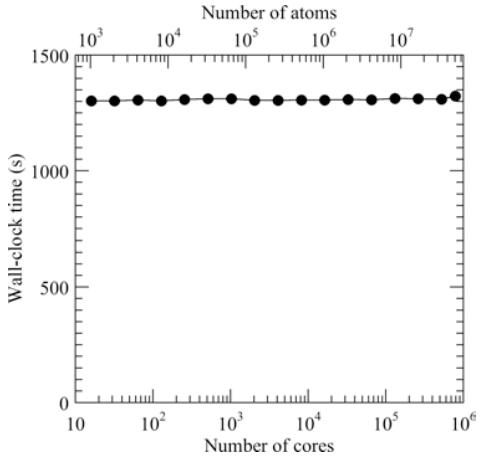

Figure 5. Wall-clock time per QMD simulation step of the parallel LDC-DFT algorithm, with scaled workloads — 64*P*-atom SiC system on *P* cores (*P* = 16, ..., 786,432) of Blue Gene/Q.

the wall-clock time remains nearly constant, indicating excellent scalability. To quantify the parallel efficiency, we first define the speed of the LDC-DFT program as a product of the total number of atoms and the number of QMD simulation steps executed per second. The isogranular speedup is given by the ratio between the speed on P cores and that on 16 cores (*i.e.* one computing node) as a reference system. The weak-scaling parallel efficiency is the isogranular speedup divided by $P/16$. With the granularity of 64 atoms per core, the parallel efficiency is 0.984 on P = 786,432 for a 50,331,648-atom (or 201,326,592-electron) SiC system. This computation involves 39,815,773,421,568 electronic wave-function and charge-density values sampled on mesh points. This result demonstrates the very high scalability of the LDC-DFT algorithm.

Next, we perform a strong-scaling test by simulating a LiAl nanoparticle immersed in water containing a total of 77,889 atoms ($Li_{2136}Al_{2136}$ immersed in 24,539 H_2O molecules). In this test, the number of cores ranges from P = 49,152 to 786,432, while keeping the total problem size constant. Figure 6 shows the wall-clock time per QMD simulation step as a function of P. The time-to-solution is reduced by a factor of 12.85 on 786,432 cores compared with the 49,152-core run (*i.e.*, using 16-times larger number of cores). This signifies a strong-scaling speedup of 12.85, with the corresponding strong-scaling parallel efficiency of 0.803.

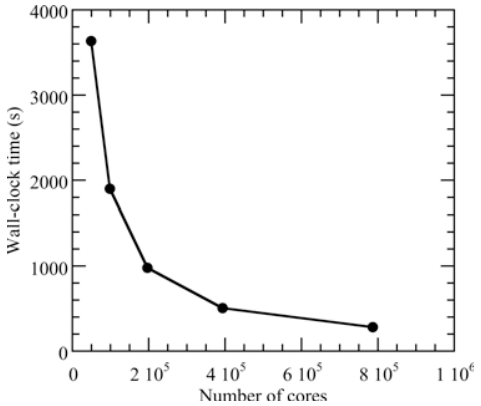

Figure 6. Wall-clock time per QMD simulation step of the parallel LDC-DFT algorithm with strong scaling — 77,889-atom LiAl-water system on *P* cores (*P* = 49,152, ..., 786,432) of Blue Gene/Q.

The excellent weak and strong scaling shown above is a consequence of minimal global communication in the DC-DFT algorithm. Namely, only one global density function $\rho(\mathbf{r})$ is shared among the DC domains instead of communicating $O(N)$ electronic wave functions $\psi_n^\alpha(\mathbf{r})$. This abstraction of global properties using reduced degrees of freedom significantly decreases the communication requirement. For the largest system involving 50.3 million atoms, the global charge density accounts for only 0.078% of the entire data size, which is far surpassed by the local electronic wave functions. In addition, the reduced buffer thickness in the new LDC-DFT algorithm (see section 5.2) has drastically reduced the point-to-point communication to exchange $\rho_\alpha(\mathbf{r})$ values among nearest neighbor domains.

5.2 Time-to-Solution

The LDC-DFT algorithm significantly reduces the $O(N)$ prefactor of its computational cost with a given energy tolerance, and consequently the time-to-solution. To quantify the reduction of the time-to-solution, we first measure the convergence of calculation with respect to the buffer thickness b, which controls the data locality of the algorithm (Fig. 1). Figure 7 shows the calculated potential energy as a function of b for an amorphous cadmium selenide (CdSe) system containing 512 atoms in a cubic simulation box of length 45.664 atomic units (a.u.). The side length l of each cubic domain is fixed as 11.416 a.u. To study the convergence of the calculated energy with respect to the localization control parameter b, we compare results of our original DC-DFT and new LDC-DFT algorithms. We see that the LDC-DFT calculation converges much more rapidly than the DC-DFT calculation. The LDC-DFT potential energy converges within 10^{-3} a.u. per atom above b = 4 a.u. In order to assess the impact of the improved convergence of LDC-DFT on the computational cost, let us consider an example of the error tolerance of 5×10^{-3} a.u. for the energy. According to Fig. 7, the buffer thickness b to achieve this convergence criterion is decreased from 4.73 a.u. for DC-DFT to 3.57 a.u. for LDC-DFT. According to the complexity analysis in section 3.1, this amounts to the computational speedup over the original algorithm by a factor of $[(11.416+2\times4.72)/(11.416+2\times3.57)]^{3\nu}$ = 2.03 (for ν = 2) or 2.89 (for ν = 3). The speedup factor of LDC-DFT over DC-DFT is a function of the energy-convergence criterion: 2.59 or 4.18 (for the energy-convergence criterion of 1×10^{-2} a.u.), 2.03 or 2.89 (5×10^{-3} a.u.), and 1.42 or 1.69 (10^{-3} a.u.) for ν = 2 or 3, respectively.

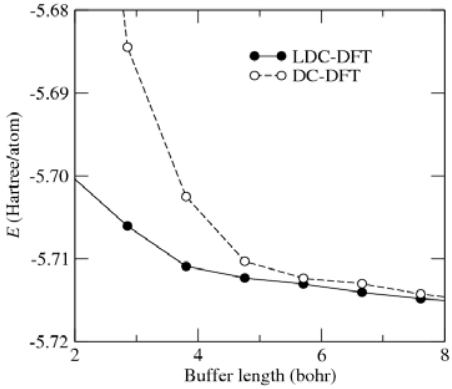

Figure 7. Energy convergence with respect to the localization control parameter. Potential energy is plotted as a function of the buffer thickness *b* for an amorphous CdSe system. The solid and open circles represent results for the LDC-DFT and original DC-DFT algorithms, respectively.

The decreased buffer thickness in LDC-DFT also reduces the crossover point, above which the $O(N)$ algorithm is faster than the conventional $O(N^3)$ DFT algorithm. The crossover point is estimated by equating the LDC-DFT computation time $T_{comp}(l_*)$ (l_* is the optimal domain length derived in section 3.1) with that of the $O(N^3)$ DFT, $L^{3\nu}$. For ν = 2, the resulting crossover point is $L = 8b$ = 28.56 a.u. for CdSe. The corresponding number of atoms is N_{atom} = 125. Even by imposing a more stringent accuracy requirement by enlarging the buffer thickness by 50%, the crossover point is 125×1.5^3 = 422 atoms, for which the above assumption (*i.e.* ν = 2) still holds [34].

To measure the actual time-to-solution, let us consider the largest computation shown in section 5.1, *i.e.*, a 50,331,648-atom SiC

system on the entire 786,432 cores of the Blue Gene/Q. One SCF iteration using LDC-DFT in this case took 441 seconds, with a speed of 114,000 atom•iteration/s. As mentioned in section 2, this represents a several orders-of-magnitude improvement over the previous state-of-the-art in terms of the time-to-solution.

5.3 Floating-Point Performance

We have measured the floating-point (FP) performance of LDC-DFT. The Blue Gene Performance Monitoring (BGPM) API is used to measure the FP performance of the entire program by linking the threaded HPM library (libmpihpm_smp).

We first study the dependence of the FP performance on the number of threads using a small number of Blue Gene/Q nodes. Here, a 512-atom SiC system is simulated by 64 MPI ranks. The number of computing nodes is varied from 4, 8 to 16 using 16, 8 and 4 MPI ranks per node, respectively. To fully utilize the double-instruction and hardware-thread units, we also vary the number of threads such that 1–4 threads are spawned on every core. Table 1 shows the measured FP operations per second (FLOP/s) along with its percentage of the theoretical peak in the parentheses. We see that the FLOP/s is an increasing function of the number of threads per core. This reflects benefits from the double-instruction units (for the case of 2 threads per core) and the hardware-thread unit (for the case of 4 threads per core). Here, we should note that saturating all hardware threads does not necessarily improve the performance. In some test cases, the FLOP/s decreases when using 4 threads per core instead of 2 threads per core. This is likely due to memory bandwidth saturation.

Table 1. Dependence of the FLOP/s performance on the number of threads on Blue Gene/Q. The number in each cell is the measured FP performance in GFLOP/s, with the corresponding percentage of the theoretical peak in a parenthesis.

Number of nodes	Number of threads per core		
	1	2	4
4	236 (28.8%)	343 (41.9%)	445 (54.3%)
8	433 (26.4%)	563 (34.4%)	746 (45.6%)
16	806 (24.6%)	1017 (31.0%)	1535 (46.8%)

Table 2. FLOP/s performance on Mira.

Number of racks	Number of nodes	Number of cores	Measured TFLOP/s	%peak FLOP/s
1	1,024	16,384	113.23	53.99
2	2,048	32,768	226.32	53.96
48	49,152	786,432	5,081.0	50.46

Next, we perform large-scale FP performance measurements. Here, 131,072-, 262,144- and 6,291,456-atom SiC systems are simulated on 1, 2 and 48 racks (or 16,384, 32,768 and 786,432 cores) of Mira, respectively. We use 16 MPI ranks per node, where each MPI rank spawns 4 threads. The benchmark results are shown in Table 2. The percentage of the peak decreases only slightly from 54% to 50.5% by increasing the number of racks from 1 to 48. The performance on the entire Mira (*i.e*. 786,432 cores) is 5.08 PFLOP/s.

5.4 Performance Portability

To study the portability of the performance to other computing architectures, we measure the FP performance on a dual Intel Xeon E5-2665 platform using a SiC crystal consisting of 64 atoms. In this benchmark, the SiC system is decomposed into eight domains and each subdomain is assigned to one MPI task. We use the Intel Fortran compiler version 14.0 for compilation, OpenMPI 1.6.4 for interprocess communication, and Intel VTune amplifier XE 2013 for the FLOP/s measurement. FFTW library version 3.3.3 and BLAS routines in Intel math kernel library (MKL) version 11.0 are linked during compilation. We perform hardware event-based sampling with the amplxe-cl command. The elapsed time and the average unhalted frequency are obtained from the CPU_CLK_UNHALTED.THREAD and CPU_CLK_UNHALTED.REF_TSC events. Since all floating-point variables are in double precision, the FLOP can be estimated with the FP_COMP_OPS_EXE.SSE_SCALAR_DOUBLE, FP_COMP_OPS_EXE.SSE_PACKED_DOUBLE and SIMD_FP_256.PACKED_DOUBLE events. Here, we assume the turbo-boosted clock for the peak performance (198 GFLOP/s per Xeon chip and 396 GFLOP/s per node). To fully utilize the hyper-threading capability, we use OMP_NUM_THREADS = 4. We have obtained 217.6 GFLOP/s (*i.e*. 55% of the peak performance), which demonstrates an excellent performance portability of the LDC-DFT code.

5.5 Verification and Validation

In the production QMD simulations presented in the next section, the DC computational parameters are chosen to achieve a convergence of the calculated energy within 10^{-3} a.u./atom. Also, the magnitude of interatomic forces is converged within 10^{-3} a.u./atom. For a direct verification of the LDC-DFT code for the problem under consideration, we have performed QMD simulations for a $Li_{30}Al_{30}$ particle in liquid water (containing a total of 606 atoms) using both $O(N)$ LDC-DFT [29] and conventional $O(N^3)$ plane-wave DFT [47, 48] codes. We have confirmed that the quantity-of-interest (*i.e*., the number of H_2 molecules produced) in these two simulations is identical. We have then used the verified LDC-DFT code to simulate larger systems. Though experimental validation of the hydrogen production rate is beyond the reach of current experimental technologies, our calculation quantitatively explains recent experimental observations [30]: (1) Hydrogen-production rate peaks at the alloy composition of equal Li and Al contents; and (2) hydrogen production is accompanied by increased pH of surrounding water.

6. SCIENCE APPLICATION FOR RENEWABLE-ENERGY

Hydrogen production from water using aluminum (Al) particles [47, 66] could provide a renewable energy cycle [67, 68]. However, its practical application is hampered by the low reaction rate and poor yield. Using the largest ever QMD simulations on 786,432 Blue Gene/Q cores based on the LDC-DFT algorithm (Fig. 8), we have shown that orders-of-magnitude faster reactions with higher yields can be achieved by alloying Al particles with lithium (Li) [30]; see also a supplementary movie, S1.mov [69]. Here, the orders-of-magnitude improvement of the time-to-solution over the previous state-of-the-art shown in section 5 has enabled QMD simulations encompassing unprecedented spatio-temporal scales, involving 16,661 atoms (or 43,708 electrons) for 21,140 time steps (or 129,208 SCF iterations) with a unit time step of 0.242 fs.

Through the analysis of the simulation data, a key nanostructural design has been identified as the abundance of neighboring Lewis acid-base pairs, where water-dissociation and hydrogen-production require very small activation energies. These reactions

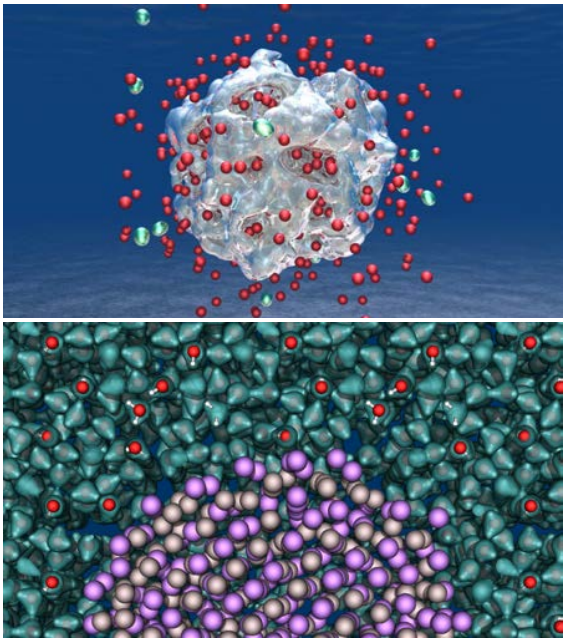

Figure 8. H_2 production from water using a $Li_{441}Al_{441}$ particle. (Top) The valance electron density (silver isosurface) is centered around Al atoms, whereas some of the Li atoms (red spheres) are dissolved into water. Green ellipsoids represent the electron charge density around produced H_2 molecules. Water molecules are not shown for clarity. (Bottom) The electron density around water molecules (cyan isosurfaces). H, O, Li and Al atoms are represented by white, red, purple and gray spheres.

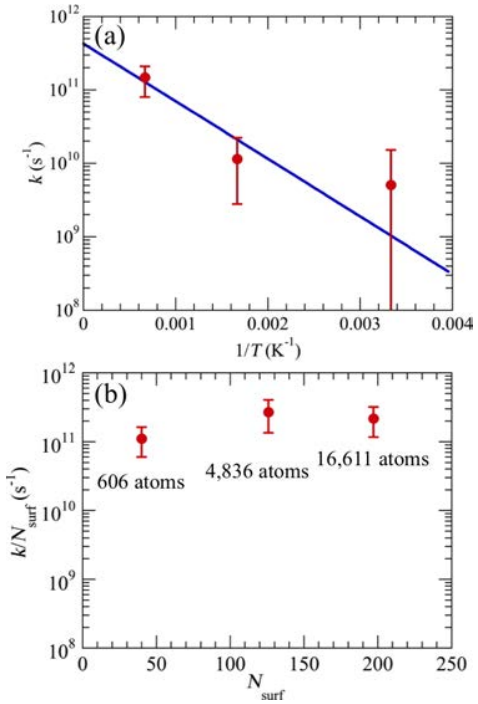

Figure 9. Rapid and scalable hydrogen-on-demand. (a) H_2 production rate as a function of inverse temperature (red circles with error bars), where the blue line is the best fit to the Arrhenius equation. (b) Hydrogen production rate normalized by the number of surface atoms, N_{surf}, as a function of N_{surf}.

are facilitated by wide charge pathways across Al atoms that collectively act as a "superanion". Furthermore, dissolution of Li atoms into water produces a corrosive basic solution that inhibits the formation of a reaction-stopping oxide layer on the particle surface, thereby increasing the yield. We have also found a surprising autocatalytic behavior of bridging oxygens (similar to Ref. [70]) that connect Al and Li. Namely, Li-O-Al is not merely an inert reaction product but instead plays an unexpectedly active role in the oxidation process by assisting the breakage of O-H and formation of Al-O bonds.

As a verification of the $O(N)$ LDC-DFT algorithm, we have first simulated a $Li_{30}Al_{30}$ particle surrounded by 182 H_2O molecules (a total of 606 atoms) at temperatures 300, 600 and 1,500 K. Figure 9(a) shows the calculated hydrogen production rate as a function of inverse temperature (an identical result was obtained by a conventional $O(N^3)$ DFT code). By the Arrhenius fit, the activation barrier is estimated to be 0.068 eV. The corresponding hydrogen production rate is 1.04×10^9 s^{-1} at 300 K per LiAl pair, which is orders-of-magnitude higher than that for pure Al [47].

A major problem of H_2 production from water using Al particles is the lack of scalability. Namely, the high reactivity of Al nanoparticles cannot be sustained for larger particles that are commercially mass-produced. To investigate the scalability of the high reactivity of Li_nAl_n with water, we have simulated larger particles, $Li_{135}Al_{135}$ and $Li_{441}Al_{441}$, in water at 1,500 K, involving a total of 4,836 and 16,611 atoms, respectively. Figure 9(b) plots the H_2 production rate normalized by the number of surface atoms, N_{surf}, as a function of N_{surf} for the three systems. The normalized H_2 production rate is constant as a function of N_{surf} within error bars. The size effect is thus negligible, indicating that the Li_nAl_n surface is equally reactive regardless of the surface curvature. Thus, the nanostructural design proposed here is expected to scale up to industrially relevant particle sizes.

Not only this microscopic understanding explains recent experimental findings in similar alloy systems (*e.g.* alloy composition-dependent reactivity and a remarkable pH change associated with H_2 production) [71], but it also predicts a specific nanostructural design for rapid high-yield production of hydrogen on demand, which is expected to scale up to industrially relevant particle sizes. This work thus lays a foundation for future studies on fundamental science toward rational nanostructural design of renewable energy technologies.

7. CONCLUSION: BROADER METASCALABLE APPLICATIONS ON FUTURE SYSTEMS

In this paper, we have presented scalable QMD simulations based on a new $O(N)$ lean divide-and-conquer density functional theory algorithm. Divide-and-conquer is a highly scalable algorithmic paradigm, which has been applied successfully to design linear-scaling algorithms for broad computational problems ranging from the formally $O(N^2)$ N-body problem [72-75], to the $O(N^3)$ eigenvalue problem [76, 77] and linear systems [78], to the exponentially complex quantum N-body problem [18-24]. With the advent of multicore revolution in computer architectures, DC software on emerging exascale computers could provide an unprecedented capability to solve complex problems. But this is

true only if the software continues to scale on the many millions of cores expected in a future exascale computer. This is an enormous challenge, since we do not even know the architecture of such platforms. The primary challenge is to sustain DC's scalability on rapidly evolving parallel computing architectures. Such a formidable challenge can only be addressed based on a solid theoretical foundation to guarantee provable scalability, adapting to evolving architectures, *i.e*., transforming the DC algorithmic framework to be metascalable [26, 31]. Light-overhead DC algorithms such as LDC-DFT are expected to be metascalable, assuming only that a tree network topology (involving progressively small communication volume at upper tree levels) will be supported. The communication requirement is further reduced algorithmically by abstracting the global information with much reduced dimensionality, *e.g*., using only one global function (density) and one scalar value (chemical potential) instead of the entire $O(N)$ wave functions in DC-DFT.

The LDC-DFT algorithm has been utilized to compute a rich variety of global properties other than mere atomic trajectories. We have recently proposed an extension of DC, which we call divide-conquer-recombine (DCR) [29]. In DCR, the DC phase computes globally informed local solutions, which are used in the recombine phase as compact bases to synthesize global properties. The recombine phase typically performs range-limited n-tuple computations [79] among DC domains to account for higher inter-domain correlations that are not captured by the tree topology used in the DC phase. For example, DC electronic wave functions have been used as compactly supported bases to calculate: (1) high-order inter-molecular-fragment interactions [80, 81]; (2) global frontier (*i.e*. highest occupied and lowest unoccupied) molecular orbitals [82, 83]; (3) global charge-migration dynamics [84-86]; and (4) global kinetics of photoexcited electrons and holes in an exciton flow network [87]; at the length and time scales that are otherwise impossible to reach. These include the largest nonadiabatic QMD simulation of photoexcited electron dynamics involving 6,400 atoms, reaching experimental spatiotemporal scales [87]. Implemented with metascalable DC algorithms such as LDC-DFT, the DCR algorithmic framework promises to scale on future systems to address many challenging scientific and engineering problems. Thus, the LDC and GSLF concepts, along with the DCR algorithmic framework, are expected to provide a metascalable computing paradigm on future systems, which is generalizable to very broad applications.

8. ACKNOWLEDGMENTS

This research was supported by the U.S. Department of Energy (DOE), Office of Science, Basic Energy Sciences, Materials Science and Engineering Division, Grant # DE-FG02-04ER-46130. We used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. DOE under contract DE-AC02-06CH11357. The initial code development and the performance portability test were conducted at the Center for High-Performance Computing and Communication at the University of Southern California. We thank Dr. Bob Walkup of IBM for his help on performance monitoring and optimization, Dr. Franz Franchetti of Carnegie Mellon University and Dr. Brian Duff of SpiralGen for their help on the use of the Spiral FFT software, and Dr. Vitali A. Morozov of ALCF for his help on debugging our code on Mira.

9. REFERENCES

[1] Car, R. and Parrinello, M., 1985. Unified approach for molecular-dynamics and density-functional theory. *Physical Review Letters 55*, 22 (Nov 25), 2471-2474.

[2] Payne, M. C., Teter, M. P., Allan, D. C., Arias, T. A., and Joannopoulos, J. D., 1992. Iterative minimization techniques for ab initio total-energy calculations - molecular-dynamics and conjugate gradients. *Reviews of Modern Physics 64*, 4 (Oct 1), 1045-1097.

[3] Rahman, A., 1964. Correlations in the motion of atoms in liquid argon. *Physical Review 136*, 2A (Oct 19), A405-A411.

[4] Hohenberg, P. and Kohn, W., 1964. Inhomogeneous electron gas. *Physical Review 136*, 3B (Nov 9), B864-B871.

[5] Kohn, W. and Sham, L. J., 1965. Self-consistent equations including exchange and correlation effects. *Physical Review 140*, 4A (Nov 15), A1133-A1138.

[6] The three most cited articles in the history of *Physical Review* journals are those on DFT; see Redner, S., 2005. *Physics Today 58*, 6 (June), 49-54.

[7] Kohn, W., 1999. Nobel lecture: electronic structure of matter-wave functions and density functionals. *Reviews of Modern Physics 71*, 5 (Oct), 1253-1266.

[8] Kikuchi, H., Kalia, R. K., Nakano, A., Vashishta, P., Iyetomi, H., Ogata, S., Kouno, T., Shimojo, F., Tsuruta, K., and Saini, S., 2002. Collaborative simulation Grid: multiscale quantum-mechanical/classical atomistic simulations on distributed PC clusters in the US and Japan. *Proceedings of Supercomputing, SC02*.

[9] Takemiya, H., Tanaka, Y., Sekiguchi, S., Ogata, S., Kalia, R. K., Nakano, A., and Vashishta, P., 2006. Sustainable adaptive Grid supercomputing: multiscale simulation of semiconductor processing across the Pacific. *Proceedings of Supercomputing, SC06*.

[10] Warshel, A. and Karplus, M., 1972. Calculation of ground and excited-state potential surfaces of conjugated molecules. 1. formulation and parametrization. *Journal of the American Chemical Society 94*, 16 (Aug), 5612-5625.

[11] Warshel, A. and Levitt, M., 1976. Theoretical studies of enzymic reactions - dielectric, electrostatic and steric stabilization of carbonium-ion in reaction of lysozyme. *Journal of Molecular Biology 103*, 2 (May 15), 227-249.

[12] Goedecker, S., 1999. Linear scaling electronic structure methods. *Reviews of Modern Physics 71*, 4 (Jul 1), 1085-1123.

[13] Bowler, D. R. and Miyazaki, T., 2012. $O(N)$ methods in electronic structure calculations. *Reports on Progress in Physics 75*, 3 (Mar), 036503.

[14] Nakano, A., Kalia, R. K., Vashishta, P., Campbell, T. J., Ogata, S., Shimojo, F., and Saini, S., 2001. Scalable atomistic simulation algorithms for materials research. *Proceedings of Supercomputing, SC01*.

[15] Kohn, W., 1996. Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters 76*, 17 (Apr 22), 3168-3171.

[16] Prodan, E. and Kohn, W., 2005. Nearsightedness of electronic matter. *Proceedings of the National Academy of Sciences 102*, 33 (Aug 16), 11635-11638.

[17] Benzi, M., Boito, P., and Razouk, N., 2013. Decay properties of spectral projectors with applications to electronic structure. *SIAM Review 55*, 1 (Feb 7), 3-64.

[18] Yang, W. T., 1991. Direct calculation of electron-density in density-functional theory. *Physical Review Letters 66*, 11 (Mar 18), 1438-1441.

[19] Dixon, S. L. and Merz, K. M., 1997. Fast, accurate semiempirical molecular orbital calculations for macromolecules. *Journal of Chemical Physics 107*, 3 (July 15), 879-893.

[20] Shimojo, F., Kalia, R. K., Nakano, A., and Vashishta, P., 2005. 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*, 3 (May 1), 151-164.

[21] Ozaki, T., 2006. *O*(*N*) Krylov-subspace method for large-scale ab initio electronic structure calculations. *Physical Review B 74*, 24 (Dec), 245101.

[22] Kobayashi, M. and Nakai, H., 2008. Extension of linear-scaling divide-and-conquer-based correlation method to coupled cluster theory with singles and doubles excitations. *Journal of Chemical Physics 129*, 4 (Jul 28), 044103.

[23] Shimojo, F., Kalia, R. K., Nakano, A., and Vashishta, P., 2008. Divide-and-conquer density functional theory on hierarchical real-space grids: parallel implementation and applications. *Physical Review B 77*, 8 (Feb 15), 085103.

[24] Ohba, N., Ogata, S., Kouno, T., Tanmura, T., and Kobayashi, R., 2012. Linear scaling algorithm of real-space density functional theory of electrons with correlated overlapping domains. *Computer Physics Communications 183*, 8 (Aug), 1664-1673.

[25] Nakano, A., Kalia, R. K., Nomura, K., Sharma, A., Vashishta, P., Shimojo, F., Van Duin, A. C. T., Goddard, W. A., Biswas, R., Srivastava, D., and Yang, L. H., 2008. De novo ultrascale atomistic simulations on high-end parallel supercomputers. *International Journal of High Performance Computing Applications 22*, 1 (Feb), 113-128.

[26] Nomura, K., Dursun, H., Seymour, R., Wang, W., Kalia, R. K., Nakano, A., Vashishta, P., Shimojo, F., and Yang, L. H., 2009. A metascalable computing framework for large spatiotemporal-scale atomistic simulations. *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS 2009*.

[27] Transferability signifies that the accuracy of an approximation holds for broad atomic configurations beyond those used in fitting. Using a large basis set as in this work is essential for achieving the transferability.

[28] Shimojo, F., Nakano, A., Kalia, R. K., and Vashishta, P., 2009. Enhanced reactivity of nanoenergetic materials: a first-principles molecular dynamics study based on divide-and-conquer density functional theory. *Applied Physics Letters 95*, 4 (Jul 27), 043114.

[29] Shimojo, F., Kalia, R. K., Kunaseth, M., Nakano, A., Nomura, K., Ohmura, S., Shimamura, K., and Vashishta, P., 2014. A divide-conquer-recombine algorithmic paradigm for multiscale materials modeling. *Journal of Chemical Physics 140*, 18 (May 14), 18A529.

[30] Shimamura, K., Shimojo, F., Kalia, R. K., Nakano, A., Nomura, K., and Vashishta, P., 2014. Hydrogen-on-demand using metallic alloy nanoparticles in water. *Nano Letters 14*, 7 (Jul 9), 4090-4096.

[31] Shimojo, F., Kalia, R. K., Nakano, A., Nomura, K., and Vashishta, P., 2008. Metascalable molecular dynamics simulation of nano-mechano-chemistry. *Journal of Physics-Condensed Matter 20*, 29 (Jul 23), 294204.

[32] Goedecker, S. and Colombo, L., 1994. Tight binding molecular dynamics. *Proceedings of Supercomputing, SC94*, 670-672.

[33] Ujfalussy, B., Wang, X., Zhang, X., Nicholson, D. M. C., Shelton, W. A., Stocks, G. M., Canning, A., Wang, Y., and Gyorffy, B. L., 1998. High performance first principles method for complex magnetic properties. *Proceedings of Supercomputing, SC98*.

[34] Gygi, F., Draeger, E., De Supinski, B. R., Yates, R. K., Franchetti, F., Kral, S., Lorenz, J., Ueberhuber, C. W., Gunneis, J. A., and Sexton, J. C., 2005. Large-scale first-principles molecular dynamics simulations on the BlueGene/L platform using the Qbox code. *Proceedings of Supercomputing, SC05*.

[35] Wang, L. W., Lee, B., Shan, H., Zhao, Z., Meza, J., Strohmaier, E., and Bailey, D. H., 2008. Linearly scaling 3D fragment method for large-scale electronic structure calculations. *Proceedings of Supercomputing, SC08*.

[36] Eisenbach, M., Zhou, C. G., Nicholson, D. M., Brown, G., Larkin, J., and Schulthess, T. C., 2009. A scalable method for ab initio computation of free energies in nanoscale systems. *Proceedings of Supercomputing, SC09*.

[37] Hasegawa, Y., Iwata, J., Tsuji, M., Takahashi, D., Oshiyama, A., Minami, K., Boku, T., Shoji, F., Uno, A., Kurokawa, M., Inoue, H., Miyoshi, I., and Yokokawa, M., 2011. First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer. *Proceedings of Supercomputing, SC11*.

[38] Kitaura, K., Ikeo, E., Asada, T., Nakano, T., and Uebayasi, M., 1999. Fragment molecular orbital method: an approximate computational method for large molecules. *Chemical Physics Letters 313*, 3-4 (Nov 12), 701-706.

[39] Ikegami, T., Ishida, T., Fedorov, D. G., Kitaura, K., Inadomi, Y., Umeda, H., Yokokawa, M., and Sekiguchi, S., 2005. Full electron calculation beyond 20,000 atoms: ground electronic state of photosynthetic proteins *Proceedings of Supercomputing, SC05*.

[40] Beran, G. J. O. and Hirata, S., 2012. Fragment and localized orbital methods in electronic structure theory. *Physical Chemistry and Chemical Physics 14*, 21 (Jun 7), 7559-7561.

[41] Osei-Kuffuor, D. and Fattebert, J.-L., 2014. Accurate and scalable *O*(*N*) algorithm for first-principles molecular-dynamics. *Physical Review Letters 112*, 4 (Jan 31), 046401.

[42] Sena, A. M. P., Miyazaki, T., and Bowler, D. R., 2011. Linear scaling constrained density functional theory in CONQUEST. *Journal of Chemical Theory and Computation 7*, 4 (Apr), 884-889.

[43] Skylaris, C. K., Haynes, P. D., Mostofi, A. A., and Payne, M. C., 2005. Introducing ONETEP: linear-scaling density functional simulations on parallel computers. *Journal of Chemical Physics 122*, 8 (Feb 22), 084119.

[44] Dziedzic, J., Hill, Q., and Skylaris, C. K., 2013. Linear-scaling calculation of Hartree-Fock exchange energy with non-orthogonal generalised Wannier functions. *Journal of Chemical Physics 139*, 21 (Dec 7), 214103.

[45] Soler, J. M., Artacho, E., Gale, J. D., Garcia, A., Junquera, J., Ordejon, P., and Sanchez-Portal, D., 2002. The SIESTA method for ab initio order-*N* materials simulation. *Journal of Physics-Condensed Matter 14*, 11 (Mar 25), 2745-2779.

[46] Duy, T. V. T. and Ozaki, T., 2014. A three-dimensional domain decomposition method for large-scale DFT electronic structure calculations. *Computer Physics Communications 185*, 3 (Mar), 777-789.

[47] Shimojo, F., Ohmura, S., Kalia, R. K., Nakano, A., and Vashishta, P., 2010. Molecular dynamics simulations of rapid hydrogen production from water using aluminum clusters as catalyzers. *Physical Review Letters 104*, 12 (Mar 26), 126102.

[48] Shimamura, K., Shimojo, F., Kalia, R. K., Nakano, A., and Vashishta, P., 2013. Bonding and structure of ceramic-ceramic interfaces. *Physical Review Letters 111*, 6 (Aug 8), 066103.

[49] Nakano, A. and Ichimaru, S., 1989. Dynamic correlations in electron liquids. 1. General formalism. *Physical Review B 39*, 8 (Mar 15), 4930-4937.

[50] Bekas, C. and Curioni, A., 2012. Algorithmic rethinking and code reengineering for truly massively parallel ab initio molecular dynamics simulations. In *Hierarchical Methods for Dynamics in Complex Molecular Systems*, J. Grotendorst, G. Sutmann, G. Gompper and D. Marx Eds. Julich Supercomputing Center, Julich, Germany, 235-268.

[51] Puschel, M., Moura, J., Johnson, J., Padua, D., Veloso, M., Singer, B., Xiong, J., Franchetti, F., Gacic, A., Voronenko, Y., Chen, K., Johnson, R. W., and Rizzolo, N., 2005. SPIRAL: code generation for DSP transforms. *Proceedings of IEEE 93*, 2 (Feb), 232-275.

[52] Frigo, M. and Johnson, S. G., 2005. The design and implementation of FFTW3. *Proceedings of IEEE 93*, 2 (Feb), 216-231.

[53] Nakano, A., Vashishta, P., and Kalia, R. K., 1994. Massively-parallel algorithms for computational nanoelectronics based on quantum molecular dynamics. *Computer Physics Communications 83*, 2-3 (Dec), 181-196.

[54] Lin, L., Lu, J. F., Ying, L. X., and E, W. N., 2012. Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: total energy calculation. *Journal of Computational Physics 231*, 4 (Feb 20), 2140-2154.

[55] Nakano, A., 2008. A space-time-ensemble parallel nudged elastic band algorithm for molecular kinetics simulation. *Computer Physics Communications 178*, 4 (Feb 15), 280-289.

[56] Vanderbilt, D., 1990. Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. *Physical Review B 41*, 11 (Apr 15), 7892-7895.

[57] Haring, R. A., Ohmacht, M., Fox, T. W., Gschwind, M. K., Satterfield, D. L., Sugavanam, K., Coteus, P. W., Heidelberger, P., Blumrich, M. A., Wisniewski, R. W., Gara, A., Chiu, G. L. T., Boyle, P. A., Christ, N. H., and Kim, C., 2012. The IBM Blue Gene/Q compute chip. *IEEE Micro 32*, 2 (Mar-Apr), 48-60.

[58] Kunaseth, M., Kalia, R. K., Nakano, A., Vashishta, P., Richards, D. F., and Glosli, J. N., 2013. Performance characteristics of hardware transactional memory for molecular dynamics application on BlueGene/Q: toward efficient multithreading strategies for large-scale scientific applications. *Proceedings of the International Workshop on Parallel and Distributed Scientific and Engineering Computing, PDSEC-13*.

[59] Chen, D., Eisley, N. A., Heidelberger, P., Senger, R. M., Sugawara, Y., Kumar, S., Salapura, V., Satterfield, D. L., Steinmacher-Burow, B., and Parker, J. J., 2012. The IBM Blue Gene/Q interconnection fabric. *IEEE Micro 32*, 1 (Jan-Feb), 32-43.

[60] Chen, D., Eisley, N., Heidelberger, P., Kumar, S., Mamidala, A., Petrini, F., Senger, R., Sugawara, Y., Walkup, R., Steinmacher-Burow, B., Choudhury, A., Sabharwal, Y., Singhal, S., and Parker, J. J., 2012. Looking under the hood of the IBM Blue Gene/Q network. *Proceedings of Supercomputing, SC12*.

[61] Gilge, M., 2013. *IBM System Blue Gene Solution Blue Gene/Q Application Development*. IBM Redbooks.

[62] Peng, L., Seymour, R., Nomura, K., Kalia, R. K., Nakano, A., Vashishta, P., Loddoch, A., Netzband, M., Volz, W. R., and Wong, C. C., 2009. High-order stencil computations on multicore clusters. *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS 2009*.

[63] https://www.alcf.anl.gov/sites/www.alcf.anl.gov/files/Hpct-bgq_0.pdf

[64] Gunnels, J. A., 2012. Making good enough...better: addressing the multiple objectives of high-performance parallel software with a mixed global-local worldview. ICERM, Brown University.

[65] Omeltchenko, A., Campbell, T. J., Kalia, R. K., Liu, X. L., Nakano, A., and Vashishta, P., 2000. Scalable I/O of large-scale molecular dynamics simulations: A data-compression algorithm. *Computer Physics Communications 131*, 1-2 (Sep 1), 78-85.

[66] Roach, P. J., Woodward, W. H., Castleman, A. W., Reber, A. C., and Khanna, S. N., 2009. Complementary active sites cause size-selective reactivity of aluminum cluster anions with water. *Science 323*, 5913 (Jan 23), 492-495.

[67] Muhich, C. L., Evanko, B. W., Weston, K. C., Lichty, P., Liang, X. H., Martinek, J., Musgrave, C. B., and Weimer, A. W., 2013. Efficient generation of H_2 by splitting water with an isothermal redox cycle. *Science 341*, 6145 (Aug 2), 540-542.

[68] Chueh, W. C., Falter, C., Abbott, M., Scipio, D., Furler, P., Haile, S. M., and Steinfeld, A., 2010. High-flux solar-driven thermochemical dissociation of CO_2 and H_2O using nonstoichiometric ceria. *Science 330*, 6012 (Dec 23), 1797-1801.

[69] http://cacs.usc.edu/download/S1.mov

[70] Wu, C. J., Fried, L. E., Yang, L. H., Goldman, N., and Bastea, S., 2009. Catalytic behaviour of dense hot water. *Nature Chemistry 1*, 1 (Apr 1), 57-62.

[71] Chen, X., Zhao, Z., Hao, M., and Wang, D., 2013. Hydrogen generation by splitting water with Al-Li alloys. *International Journal of Energy Research 37*, 13 (Oct 25), 1624-1634.

[72] Greengard, L. and Rokhlin, V., 1987. A fast algorithm for particle simulations. *Journal of Computational Physics 73*, 2 (Dec), 325-348.

[73] White, C. A. and Headgordon, M., 1994. Derivation and efficient implementation of the fast multipole method. *Journal of Chemical Physics 101*, 8 (Oct 15), 6593-6605.

[74] Nakano, A., Kalia, R. K., and Vashishta, P., 1994. Multiresolution molecular-dynamics algorithm for realistic materials modeling on parallel computers. *Computer Physics Communications 83*, 2-3 (Dec), 197-214.

[75] Ogata, S., Campbell, T. J., Kalia, R. K., Nakano, A., Vashishta, P., and Vemparala, S., 2003. Scalable and portable implementation of the fast multipole method on parallel computers. *Computer Physics Communications 153*, 3 (Jul 1), 445-461.

[76] Cuppen, J. J. M., 1981. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik 36*, 2, 177-195.

[77] Gansterer, W. N., Ward, R. C., and Muller, R. P., 2002. An extension of the divide-and-conquer method for a class of symmetric block-tridiagonal eigenproblems. *ACM Transactions on Mathematical Software 28*, 1 (Mar), 45-58.

[78] Nakano, A., 1997. Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. *Computer Physics Communications 104*, 1-3 (Aug), 59-69.

[79] Kunaseth, M., Kalia, R. K., Nakano, A., Nomura, K., and Vashishta, P., 2013. A scalable parallel algorithm for dynamic range-limited *n*-tuple computation in many-body molecular dynamics simulation. *Proceedings of Supercomputing, SC13*.

[80] Okiyama, Y., Tsukamoto, T., Watanabe, C., Fukuzawa, K., Tanaka, S., and Mochizuki, Y., 2013. Modeling of peptide-silica interaction based on four-body corrected fragment molecular orbital (FMO4) calculations. *Chemical Physics Letters 566* (Apr 12), 25-31.

[81] Tanaka, S., Watanabe, C., and Okiyama, Y., 2013. Statistical correction to effective interactions in the fragment molecular orbital method. *Chemical Physics Letters 556* (Jan 29), 272-277.

[82] Tsuneyuki, S., Kobori, T., Akagi, K., Sodeyama, K., Terakura, K., and Fukuyama, H., 2009. Molecular orbital calculation of biomolecules with fragment molecular orbitals. *Chemical Physics Letters 476*, 1-3 (Jul 7), 104-108.

[83] Kobori, T., Sodeyama, K., Otsuka, T., Tateyama, Y., and Tsuneyuki, S., 2013. Trimer effects in fragment molecular orbital-linear combination of molecular orbitals calculation of one-electron orbitals for biomolecules. *Journal of Chemical Physics 139*, 9 (Sep 7), 094113.

[84] Gollub, C., Avdoshenko, S., Gutierrez, R., Berlin, Y., and Cuniberti, G., 2012. Charge migration in organic materials: can propagating charges affect the key physical quantities controlling their motion? *Israel Journal of Chemistry 52*, 5 (May), 452-460.

[85] Kitoh-Nishioka, H. and Ando, K., 2012. Fragment molecular orbital study on electron tunneling mechanisms in bacterial photosynthetic reaction center. *Journal of Physical Chemistry B 116*, 43 (Nov 1), 12933-12945.

[86] Ren, J. F., Vukmirovic, N., and Wang, L. W., 2013. Nonadiabatic molecular dynamics simulation for carrier transport in a pentathiophene butyric acid monolayer. *Physical Review B 87*, 20 (May 13), 205117.

[87] Mou, W., Hattori, S., Rajak, P., Shimojo, F., and Nakano, A., 2013. Nanoscopic mechanisms of singlet fission in amorphous molecular solid. *Applied Physics Letters 102*, 17 (Apr 29), 173301.