The midpoint method for parallelization of particle simulations

Kevin J. Bowers and Ron O. Dror
D. E. Shaw Research, LLC, 39th Floor, Tower 45, 120 West 45th Street, New York, New York 10036

David E. Shaw[a)]
D. E. Shaw Research, LLC, 39th Floor, Tower 45, 120 West 45th Street, New York, New York 10036 and Center for Computational Biology and Bioinformatics, Columbia University, 1130 St. Nicholas Avenue, New York, New York 10032

(Received 17 January 2006; accepted 8 March 2006; published online 12 May 2006)

The evaluation of interactions between nearby particles constitutes the majority of the computational workload involved in classical molecular dynamics (MD) simulations. In this paper, we introduce a new method for the parallelization of range-limited particle interactions that proves particularly suitable to MD applications. Because it applies not only to pairwise interactions but also to interactions involving three or more particles, the method can be used for evaluation of both nonbonded and bonded forces in a MD simulation. It requires less interprocessor data transfer than traditional spatial decomposition methods at all but the lowest levels of parallelism. It gains an additional practical advantage in certain commonly used interprocessor communication networks by distributing the communication burden more evenly across network links and by decreasing the associated latency. When used to parallelize MD, it further reduces communication requirements by allowing the computations associated with short-range nonbonded interactions, long-range electrostatics, bonded interactions, and particle migration to use much of the same communicated data. We also introduce certain variants of this method that can significantly improve the balance of computational load across processors. © *2006 American Institute of Physics*.
[DOI: 10.1063/1.2191489]

I. INTRODUCTION

Molecular dynamics (MD) simulations of biological macromolecules in explicit solvent aim to provide a computational "microscope," yielding insight into biochemical events that are difficult to observe experimentally.[1] Simulations of significantly more than a microsecond, however, require tremendous computational power and will likely only become practical through the use of a large number of processors in parallel.[2] Such parallelism is limited by interprocessor communication requirements. In this paper, we introduce the *midpoint method*, a parallelization method that proves particularly useful for MD.

Most of the computational workload in MD is associated with the evaluation of electrostatic and van der Waals forces between all pairs of atoms separated by less than some *interaction radius R*.[3] The explicit evaluation of interactions between pairs of particles separated by less than some maximum distance also constitutes the majority of the computational expense of gravitational simulations in astrophysics, particle simulations in plasma physics, and smooth particle hydrodynamic simulations in fluid dynamics.[4,5] Traditional methods for parallelizing such problems, described in several review papers,[6] include atom, force, and spatial decomposition methods. Unlike atom and force decomposition methods, spatial decomposition methods offer the desirable property that the amount of data to be transferred into and out of each processor (the method's *communication bandwidth*) decreases as the interaction radius decreases.

A number of recently introduced methods for parallelizing range-limited particle interactions[7–9] require significantly less communication bandwidth than traditional parallelization methods. These novel methods are similar to traditional spatial decomposition methods in that each processor takes responsibility for updating positions of particles that fall in a certain region of space and in that their communication bandwidth decreases as the interaction radius decreases. Unlike traditional spatial decomposition methods, however, these new methods sometimes compute the interaction between a pair of particles on a processor on which neither particle resides.[8] We refer to such techniques as *neutral territory* methods.

The midpoint method is a neutral territory method that requires less communication bandwidth than any traditional spatial decomposition method at all but extremely low levels of parallelism, where it requires the same amount of communication bandwidth as traditional methods. While certain other neutral territory methods require less communication bandwidth than the midpoint method for pairwise interactions parallelized over a sufficiently large number of processors, the midpoint method offers significant advantages over previously described methods in several respects. It applies not only to pairwise interactions, but also to interactions involving sets of three or more particles. In addition, it typically incurs a smaller penalty due to communication latency than other methods. It also communicates equal amounts of

[a)]Author to whom correspondence should be addressed. Electronic mail: david@deshaw.com

data in each direction, leading to more effective use of communication links in certain network topologies.

The midpoint method enjoys additional advantages in the context of MD computations. It can be used to parallelize the evaluation of bonded force terms, which commonly involve the interaction of up to four particles, as well as pairwise electrostatic and van der Waals terms. Moreover, it allows several components of the MD computation, including the evaluation of the bonded force terms, to rely on the same data that are communicated for the evaluation of electrostatic and van der Waals forces between pairs of nearby atoms. The midpoint method thus proves a particularly attractive choice for the parallelization of MD computations on up to at least 512 processors. (Throughout this paper, we use the term "processor" to refer to a processing node connected to other such nodes through a communication network; one such processor might actually include a number of independent processing units, but we assume that the communication between them is faster than the communication between processing nodes, and thus focus on parallelization across, but not within, the processing nodes.) Recent IBM technical reports describe an independently developed but related technique for parallelizing MD on Blue Gene/L.[10,11]

We also describe certain variants of the midpoint method, to which we refer as *midpoint-ensured methods*, that lend themselves particularly well to load balancing. In a midpoint-ensured method, each processor imports the same data as would be imported in the midpoint method, but the assignment of interaction computations to processors is adjusted in such a way that the computational load on different processors is more even.

II. SPECIFICATION OF THE MIDPOINT METHOD

In the midpoint method, as in traditional spatial decomposition methods and previously described neutral territory methods, each processor assumes responsibility for updating the positions of particles in a distinct region of space, regardless of where the particle interactions are computed. When applied to pairwise interactions, the midpoint method specifies that two particles interact on a particular processor if and only if the midpoint of the segment connecting them falls within the region of space associated with that processor.

We refer to the region containing the system to be simulated as the *global cell*. One can use the midpoint method with a global cell of any shape and with any regular or irregular partition of the global cell into regions assigned to different processors.[12] To simplify our exposition, we will assume in this paper that the global cell is a rectangular parallelepiped of dimensions $G_x \times G_y \times G_z$ and that it is divided into a regular grid of smaller rectangular parallelepipeds that we call *boxes*. Each processor updates the coordinates of particles in one box, referred to as the *home box* of that processor and of those particles. In the interest of simplicity, we will refer interchangeably to a processor and its home box. We refer to particles that do not reside in a particular box as *remote* to that box. When working in a three-dimensional space, we denote the dimensions of each box by $b_x \times b_y \times b_z$, and we refer to the quantities b_x/b_y and b_x/b_z as the *box aspect ratios*. The *base coordinates* of a given box and of any particle located within that box are defined as the coordinates of the low-coordinate corner of that box.

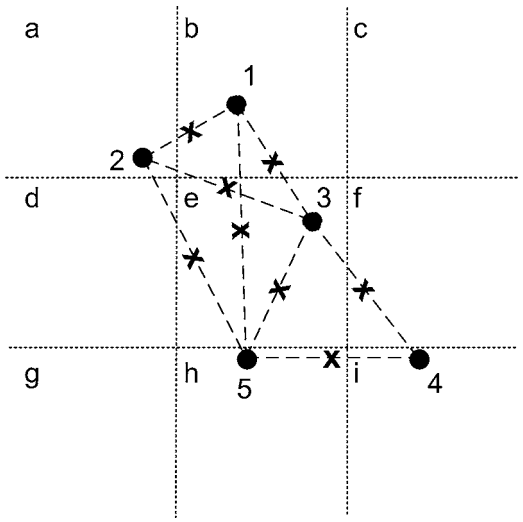


FIG. 1. Assignment of particle pairs to interaction boxes in the midpoint method. In this figure, the boxes are square with side length b and $R = 1.5b$. Each pair of particles separated by a distance less than R is connected by a dashed line segment, with the "x" at its center lying in the box which will compute the interaction of that pair.

We will assume that the global cell tiles an infinite space by repeating in each dimension with a period equal to the side length of the global cell in that dimension. The periodic boundary conditions imposed by this assumption simplify our exposition, but the methods discussed in this paper are also applicable to systems with other boundary conditions. We will also assume for simplicity that $G_x \geqslant b_x + 2R$, $G_y \geqslant b_y + 2R$, and $G_z \geqslant b_z + 2R$, so that at most one image of a given particle interacts with particles in a given box.

We refer to the box in which a set of particles interact as their *interaction box*. Figure 1 illustrates the assignment of particle pairs to interaction boxes implied by the midpoint method. Two particles that lie in the same box necessarily interact in that box, but particles that lie in different boxes may interact either in the box in which one of them resides (e.g., particles 2 and 3) or in a box in which neither resides (e.g., particles 1 and 5 or particles 3 and 4).

If two particles are separated by a distance $r < R$, then the distance from either particle to their midpoint is $r/2 < R/2$. If the midpoint lies within a particular box, then each particle must lie either within that box or less than a distance $R/2$ from its boundary. In the midpoint method, the volume of space from which a given processor must "import" particle data that ordinarily resides within other processors (its *import region*) therefore includes only points within a distance $R/2$ of its home box. This import region is shown in Fig. 2(a) for a two-dimensional system and in Fig. 3(a) for a three-dimensional system. In a MD simulation, or any other application that requires computation of the total force on each particle, each interaction box must "export" a force contribution to each of the particles in its import region after it

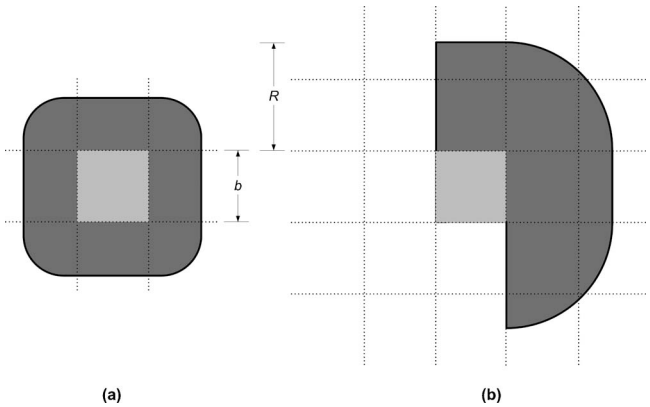


FIG. 2. Import regions of (a) the two-dimensional midpoint method and (b) the two-dimensional analog of the HS method. The boxes are square with side length b and $R=1.5b$. In each case, the interaction box is shown in light gray and the import region in dark gray.

has computed all the interactions assigned to it. For all of the generalized spatial decomposition methods described in this paper, the volume of space to which a given processor must export force contributions is identical to its import region. In applications that require only computation of a global potential energy, force export is not required.

Figures 2 and 3 also show the import regions of several previously described methods. The HS method is an example of a traditional spatial decomposition method, in which the box that interacts two particles is always the home box of one or both of them. In the HS method, the particles interact

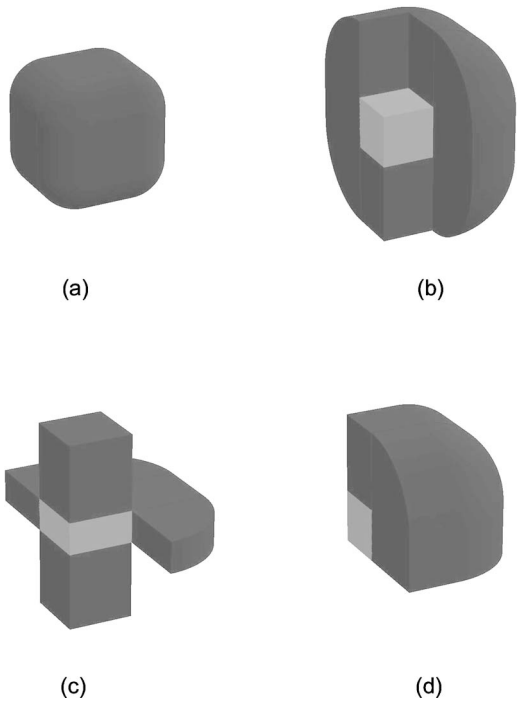


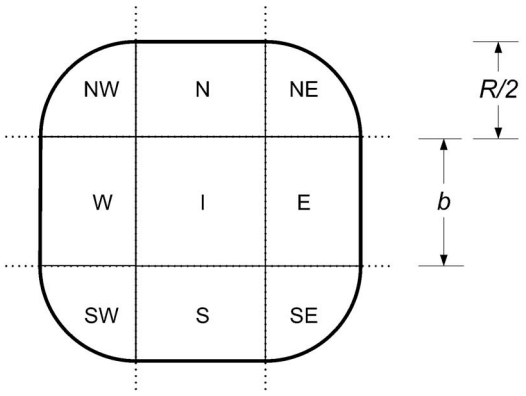
FIG. 3. Import regions of (a) the three-dimensional midpoint method, (b) the HS method, (c) the NT method, and (d) the ES method. In each case, the interaction box is shown in light gray and the import region in dark gray. The diagrams of the midpoint method, the HS method, and the ES method assume that the boxes are cubic with side length b and that $R=1.5b$. The diagram of the NT method assumes the same values for R and for box volume, but box aspect ratios have been optimized to minimize import volume.

within the home box of the particle with the smaller x base coordinate, with ties broken first in favor of the particle with the smaller y base coordinate and then in favor of the particle with the smaller z base coordinate. Figure 3(b) shows the import region of the HS method, and Fig. 2(b) shows the import region of a two-dimensional analog of that method.

The NT and ES methods, whose import regions are shown in Figs. 3(c) and 3(d), are neutral territory methods that we recently introduced.[7–9] In the NT method, two particles interact within a box that has the x and y base coordinates of one particle and the z base coordinate of the other particle. Specifically, the x and y base coordinates of the interaction box are those of the particle with the smaller x base coordinate, with ties broken first in favor of the particle with the smaller y base coordinate and then in favor of the particle with the larger z base coordinate. The z base coordinate of the interaction box is the z base coordinate of the other particle. In the ES method, a pair of particles interacts within a box whose x base coordinate is that of the particle with the smaller x coordinate, whose y base coordinate is that of the particle with the smaller y coordinate, and whose z base coordinate is that of the particle with the smaller z coordinate.

In a previous paper,[8] we described the *zonal methods*, a class of parallelization algorithms that include both traditional spatial decomposition methods (e.g., the HS method) and certain neutral territory methods (e.g., the NT and ES methods). In a zonal method, one associates with each box a set of regions called *zones*, where each box has the same spatial relationship with its zones as does each other box. Each box "interacts" certain pairs of its zones by computing the interaction between each particle in one zone and each particle in the other zone. This procedure may lead to some duplicate interactions or to interactions between pairs of particles separated by a distance greater than R, but such redundant interactions can be avoided if each processor computes interactions only for particle pairs that satisfy certain test criteria. The midpoint method for pairwise particle interactions can also be formulated as a zonal method, providing a convenient means to implement the midpoint method. Figure 4 shows a formulation with nine zones for the midpoint method in two dimensions; an analogous formulation in three dimensions would require 27 zones. Alternatively, one might divide the import region and the interaction box into a larger number of zones, which may result in higher performance by reducing wasted computation associated with particle pairs that fail the test criteria and by providing more opportunities to overlap communication time with computation time.

The midpoint method can also handle interactions that involve sets of three or more particles. Suppose that we wish to compute interactions between all sets of m particles that fall within some sphere of radius $R/2$ (if $m=2$, this is equivalent to the statement that we wish to compute interactions between all pairs of particles separated by less than a distance R). In the midpoint method, the interaction between a set of m particles is computed on the processor that contains their midpoint, defined as the center of the smallest sphere



(a)

	I	E	S	W	N	NE	SE	SW	NW
I	1	1	1	1	1	1	1	1	1
E		0	1	1	1	0	0	1	1
S			0	1	1	1	0	0	1
W				0	1	1	1	0	0
N					0	0	1	1	0
NE						0	0	1	0
SE							0	0	1
SW								0	0
NW									0

(b)

FIG. 4. A formulation of the two-dimensional midpoint method as a zonal method. (a) The interaction box (I) represents a single zone, and the import region is partitioned into eight zones. (b) An *interaction schedule* indicating which pairs of zones interact. An entry of 0 indicates that the zones in the corresponding row and column do not interact, while an entry of 1 indicates that they do.

that contains all m particles. The import region of the midpoint method in the m-particle case is identical to that for the two-particle case.

The import region of the HS method also guarantees that at least one processor will import all the particles necessary to compute the interaction between any set of m particles that fall within a sphere of radius $R/2$, but this is not the case for previously described neutral territory methods such as the NT and ES methods. The ES method does generalize in a straightforward manner to interactions of m particles; it assigns the interaction of a set of particles to the box whose x base coordinate is that of the particle with the smallest x coordinate, whose y base coordinate is that of the particle with the smallest y coordinate, and whose z base coordinate is that of the particle with the smallest z coordinate. This generalization requires expansion of the import region, however, as discussed further in the Appendix.

In practice, determining the midpoint of three or more particles involves significantly more computation than determining the midpoint of two particles. One can reduce the required amount of computation at the expense of a small amount of additional communication by employing the midpoint-ensured methods introduced in Sec. IV or by using an approximation to the midpoint, as described in the Appendix.

III. PERFORMANCE CHARACTERISTICS OF THE MIDPOINT METHOD

A. Communication bandwidth

Assuming uniform particle density, the amount of particle data that must be transferred into each processor during particle import (and out of each processor during force export, in applications that require force calculation) is proportional to the volume of the import region. We will therefore use the volume of the import region (V_i) as a measure of communication bandwidth requirements. This volume depends not only on the interaction radius and the volume of each box (V_b), but also on the box aspect ratios. For the midpoint, HS, and ES methods, one minimizes the import volume for a fixed interaction radius and box volume by using cubic boxes. For the NT method, on the other hand, noncubic boxes typically minimize the import volume.[9]

For the parallelization methods discussed in this paper, the shape of the import region depends only on the ratios of the box side lengths to the interaction radius R. In three dimensions, the ratio of the import volume to the box volume (V_i/V_b) for a particular method can be determined uniquely given R/b_x, R/b_y, and R/b_z. Alternatively, V_i/V_b can be expressed as a function of the box aspect ratios b_x/b_y and b_x/b_z and the *parallelization parameter* α_R, where we define α_R as the geometric mean of R/b_x, R/b_y, and R/b_z as follows:

$$\alpha_R = \left(\frac{R}{b_x}\frac{R}{b_y}\frac{R}{b_z}\right)^{1/3} = \left(\frac{R^3}{V_b}\right)^{1/3} = \frac{R}{V_b^{1/3}}.$$

For cubic boxes of side length b, α_R is simply R/b. The parallelization parameter α_R might be viewed as a measure of the extent to which a particular simulation has been parallelized.

Assuming cubic boxes, the import volume for the HS method is

$$V_i = V_b\left(3\alpha_R + \tfrac{3}{2}\pi\alpha_R^2 + \tfrac{2}{3}\pi\alpha_R^3\right),$$

while that for the midpoint method is

$$V_i = V_b\left(3\alpha_R + \tfrac{3}{4}\pi\alpha_R^2 + \tfrac{1}{6}\pi\alpha_R^3\right).$$

The import volume of the midpoint method is always smaller than that of the HS method, with the difference growing in both relative and absolute terms as α_R grows. The import volume of the ES method for pairwise interactions[8] is identical to that of the midpoint method. To the best of our knowledge, no published method has a smaller import volume than the midpoint method for small α_R (i.e., at low degrees of parallelism).[13]

Asymptotically, the import volumes of the midpoint, ES, and HS methods all grow with the cube of α_R, while the import volumes of certain other neutral territory methods, including the NT method, grow as $\alpha_R^{3/2}$.[8,9,14] Thus for large values of α_R, the NT method has a lower import volume than the midpoint or ES methods. More specifically, the NT method has a lower import volume than the midpoint method when $\alpha_R > 0.82$, assuming box aspect ratios are chosen to minimize the import volume for each method.

Figure 5 shows the import volumes[15] of the various parallelization methods as a function of the number of proces-

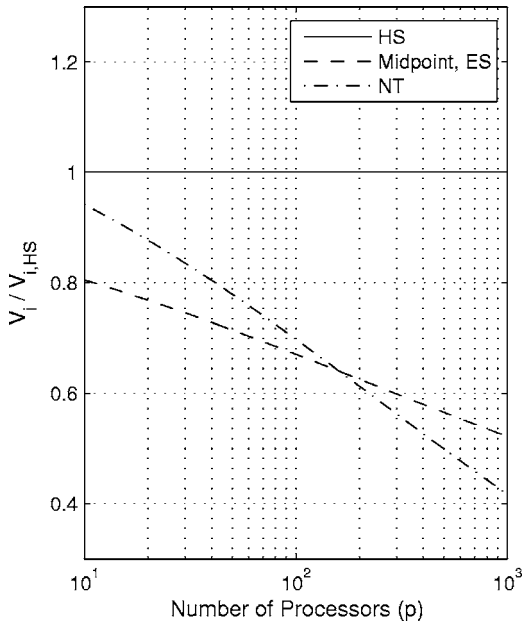


FIG. 5. Import volumes of several parallelization methods for pairwise interactions as a function of the number of processors p, assuming a cubic global cell with a side length of 80 Å and an interaction radius $R=12$ Å. Import volumes are represented relative to that of the HS method ($V_{i,\mathrm{HS}}$) for any number of processors. The box aspect ratios of the NT method were optimized at each value of p to minimize import volume, while the boxes are assumed to be cubic for the other methods. The midpoint and ES methods have the same import volume for all p.

sors for a system with a cubic global cell measuring 80 Å on a side and an interaction radius of 12 Å. These parameters are within the range that might be typical for the computation of explicit pairwise electrostatic and van der Waals forces in a MD simulation of a biomolecular system; at a typical density of 0.1 atoms/Å^3, such a system would contain about 51 000 atoms. The midpoint and ES methods have the lowest import volume of the methods shown for up to 160 processors, with the NT method achieving a lower import volume for higher numbers of processors. The kZ-NT method, a variant of the NT method that we described in a recent paper,[8] has a slightly lower import volume than any of the methods shown here for more than 63 processors. As discussed in the remainder of the paper, however, the midpoint method has a number of advantages over the competing methods that are not captured by the import volume.

The one situation in which the midpoint method requires the same import volume as traditional spatial decompositions, rather than a smaller one, is when the global cell is partitioned into boxes along only one dimension, that is, when the grid of boxes has dimensions $1\times1\times n$, $1\times n\times1$, or $n\times1\times1$. Such a situation violates our assumption that $G_x\geqslant b_x+2R$, $G_y\geqslant b_y+2R$, and $G_z\geqslant b_z+2R$ (Sec. II); the midpoint method still applies in this case, as do traditional spatial decompositions, but their import volumes are different from those computed above, as communication is only required along one dimension. A one-dimensional partition of the global cell maximizes the bandwidth only at very low levels of parallelism. In our example system above, such an approach has minimal import volume only when using fewer than six processors. The midpoint method also applies when the global cell is partitioned along two out of three dimensions (e.g., a grid of boxes with dimensions $1\times n_1\times n_2$, where $n_1\geqslant2$ and $n_2\geqslant2$); in this situation, it always has a lower import volume than traditional spatial decompositions.

Thus far we have concentrated our analysis on import volume, which serves as an accurate proxy for per processor communication bandwidth when particles are distributed uniformly and when a single interaction radius R is used throughout the simulation, as is generally the case for explicit solvent molecular dynamics simulations. When these assumptions do not hold, different processors may need to import different amounts of data, which will negatively impact the scalability of any of the methods discussed in this paper.

In a system with multiple communication links, communication will typically be most efficient if the communication load is spread evenly over all the links. The symmetry of the midpoint method's import region therefore proves advantageous for certain communication network topologies such as the three-dimensional toroidal mesh used in IBM's BlueGene/L, the Sandia/Cray Red Storm, Cray's T3D, T3E, and XT3, and some LINUX clusters.[16] In this topology each processor is connected to the processors adjacent to it in each of the six cardinal directions $(+x,-x,+y,-y,+z,-z)$. For three-dimensional simulation, the midpoint, HS, ES, and NT methods all map naturally to this topology, but the midpoint method has an advantage over the others because it transfers an approximately equal amount of data in each of the six directions. The ES method, on the other hand, communicates data only in the $-x$, $-y$, and $-z$ directions during particle import, so while the ES and midpoint methods have the same import volumes, the maximum bandwidth per communication link for the ES method will be roughly double that for the midpoint method in such an architecture. The import procedures of the NT and HS methods communicate data in five of the six cardinal directions, but they send nearly twice as much data in the $-x$ direction as in any other direction.

B. Communication latency

For certain network topologies and protocols, the midpoint method gains a further advantage from the fact that the maximum distance of a point in the import region from the interaction box boundary is smaller for the midpoint method than for other published methods. This is perhaps most obvious for a toroidal mesh network, where the *communication latency* associated with a pair of processors (a fixed time interval associated with the overhead entailed in transferring a message from one processor to another, independent of the amount of data contained in that message) increases with the *hop count*, the number of interprocessor connections a message must traverse to get from one processor to the other. In such a network, the maximum hop count and therefore the maximum communication latency associated with particle import under the midpoint method are less than or equal to that under the HS or ES method. At high levels of parallelism, the hop count of the midpoint method will be approximately half that of the HS and ES methods. For example, consider a three-dimensional case with cubic boxes and with

$\alpha_R=2$ (i.e., $R=2b$, where b is the box side length). The midpoint method requires each interaction box to import data from the boxes with which it shares a face, edge, or corner. Communication between boxes that share a face requires one hop, communication between boxes that share an edge but no face requires two hops, and communication between boxes that share only a corner requires three hops, so the maximum hop count for the midpoint method in this case is three. The HS and ES methods, on the other hand, have a maximum hop count of six in this case, because these methods require the import of data from more distant boxes in every direction. The maximum hop count for the NT method in this case is four, whether one constrains the boxes to be cubic or optimizes their aspect ratios to minimize the import volume. At low levels of parallelism, the maximum hop count of the NT method may be less than that of the midpoint method; in particular, if R was less than any box side length, the maximum hop count of the NT method would be two, while that of the midpoint method would still be three. The maximum hop count of the NT method is always greater than or equal to that of the midpoint method when $\alpha_R>\sqrt[3]{2}$, or when boxes are cubic and $\alpha_R>1$.

In many communication networks, the time required by a processor to send data depends not only on the amount of data but also on the number of separate messages to be sent. As a result, the number of other processors with which each processor must communicate directly has a significant impact on total communication time. Whenever $\alpha_R>1$, the midpoint method will require each processor to send fewer messages than the NT and HS methods and the same number of messages as the ES method, whether the box aspect ratio is constrained to be cubic or adjusted to minimize the import volume. Consider again the three-dimensional case with cubic boxes and $\alpha_R=2$ (i.e., $R=2b$). The midpoint method requires that each processor import data from its 26 nearest neighbors. We can accomplish all communication necessary for particle import in six send/receive cycles by first sending all data that must move in the $+x$ direction one step in that direction, then sending all data that must move in the $-x$ direction one step in that direction, and then performing similar send operations for the $+y$, $-y$, $+z$, and $-z$ directions. The ES method also requires six send/receive cycles, because data must move up to two steps in each of the $-x$, $-y$, and $-z$ directions. The NT and HS methods require ten send/receive cycles, however, because data must move up to two steps in each of the $-x$, $+y$, $-y$, $+z$, and $-z$ directions.

C. Further advantages for molecular dynamics simulations

The midpoint method proves particularly suitable for parallelization of MD simulations because it allows several different components of the computation to utilize the same communicated particle data, thereby resulting in lesser total communication time than if each component were parallelized separately.

The total force on a particle in most common biomolecular force fields, such as CHARMM,[17] AMBER,[18] OPLS-AA,[19] GROMOS,[20] and MMFF,[21] is expressed as a sum of bonded force terms, which depend on the covalent bond structure of the molecules, and nonbonded force terms, which comprise electrostatic and van der Waals interactions between all pairs of particles in the system. Bonded force terms include bond length terms, involving two particles connected by a bond; bond angle terms, involving three particles, two of which are bonded to a third; and dihedral angle (torsional) terms, involving four particles connected by three bonds. We denote by r_{bonded} the maximum radius of a sphere required to enclose the particles involved in any one bonded force term. In common biomolecular force fields, r_{bonded} is no larger than 4 Å.

The nonbonded force terms involve all pairs of particles in the system and therefore represent a much larger computational burden than the bonded force terms. The van der Waals forces fall off sufficiently quickly with distance that they can typically be neglected for pairs of particles separated by more than some cutoff distance $d_{\text{nonbonded}}$, typically chosen between 9 and 12 Å. An evermounting body of evidence shows that neglecting electrostatic interactions beyond a cutoff is inadequate for explicit solvent simulations,[22] so electrostatic forces are typically computed by one of several efficient, approximate methods that take long-range interactions into account without requiring the explicit interaction of all pairs of particles. The most common of these are mesh-based Ewald methods such as particle-particle particle mesh (PPPM),[4] particle mesh Ewald (PME),[23] lattice Gaussian multigrid (LGM),[24] and Gaussian split Ewald (GSE),[25] which use the fast Fourier transform (FFT) or a multigrid solver to compute electrostatic potential on a mesh given a charge distribution on that mesh. These methods require that modified electrostatic interactions be computed explicitly within some cutoff radius; we will assume that this cutoff radius is chosen to be identical to the van der Waals cutoff distance $d_{\text{nonbonded}}$, as is typically the case. These methods also require that charge be mapped from particles to nearby mesh points before the FFT or multigrid computation, and that forces on particles be calculated from potentials at nearby mesh points afterwards. In the *charge spreading* computation, the charge on each mesh point is determined as a sum over nearby particles, while in the *force interpolation* computation, the force on each particle is computed as a sum over nearby mesh points. We denote by $d_{\text{spreading}}$ the maximum distance between any particle and any mesh point that "interact" in either charge spreading or force interpolation. The value of $d_{\text{spreading}}$ depends on the parameters of the long-range electrostatics method employed, but typical values are around 5 Å.

When one parallelizes the computation of the explicit nonbonded force terms using the midpoint method, each processor must import data for particles lying within a distance $r_{\text{nonbonded}}=d_{\text{nonbonded}}/2$ from the boundary of its home box. Once it has completed computation of pairwise interactions, it must export a force contribution for each of these particles to the respective particle's home box. Because $d_{\text{nonbonded}}$ typically lies between 9 and 12 Å, $r_{\text{nonbonded}}$ typically lies between 4.5 and 6 Å.

If $d_{\text{spreading}}\leqslant r_{\text{nonbonded}}$, then no additional communication is required to support charge spreading and force interpola-

tion. The import requirements of the explicit nonbonded force computation ensure that each processor will import all remote particles within a distance $r_{\text{nonbonded}}$ of any mesh point lying in the home box. Likewise, the export requirements of the explicit nonbonded force computation ensure that each processor will export force contributions to this set of remote particles. The force contribution on each remote particle due to mesh points in the home box can be combined with the corresponding explicit nonbonded force contribution before it is exported. In MD runs using common biomolecular force fields, $d_{\text{spreading}}$ is usually approximately equal to $r_{\text{nonbonded}}$. Parameters of the mesh-based Ewald method employed can be tuned to guarantee that $d_{\text{spreading}} \leqslant r_{\text{nonbonded}}$; alternatively, the import region can be expanded slightly to include all particles within a distance $d_{\text{spreading}}$ of the home box boundary.

The computation of bonded force terms can also be parallelized using the midpoint method, which specifies that each bonded term be computed by the processor that contains the midpoint of the particles involved in that term. If $r_{\text{bonded}} \leqslant r_{\text{nonbonded}}$—as is typically the case for MD simulations using common biomolecular force fields—then no additional communication is required to support the computation of the bonded terms; the import requirements of the explicit nonbonded force computation ensure that data for each particle will be available on any processor that needs it for bonded force term computation, while the export requirements ensure that a force contribution will be exported back to the processor on which the particle resides.

In a MD simulation parallelized via the midpoint method, particles migrate from one box to another as they move. As long as the maximum distance d_{motion} that a particle can move in a time step—typically under 0.1 Å—is less than $r_{\text{nonbonded}}$, no extra communication will be required to migrate the particle position data. These data will be communicated as part of the import data required by the midpoint method for explicit nonbonded interactions (in certain cases these data will need to be imported from more boxes than if particle migration had been performed separately; because migration at the end of a time step is delayed to coincide with particle import during the following time step, data for particles within a distance $r_{\text{nonbonded}}$ of a given interaction box may still reside on the home boxes of points within a distance $r_{\text{nonbonded}}+d_{\text{motion}}$ away when import is performed). If one uses the NT, HS, or ES methods, on the other hand, particle migration will require additional communication of particle positions, because proximity of a particle to an interaction box does not guarantee that the particle will be imported by that interaction box. In some molecular dynamics implementations, particle migration may also require communication of particle state variables other than position, such as velocity, but the midpoint method will still require less total communication for particle migration than the NT, HS, or ES method. Moreover, in the midpoint method, any additional data required for migration can be included in the same communication step as the particle position import, avoiding the need for an additional round of communication.

Consider a MD simulation with a cubic global cell measuring 80 Å on a side, running on a system with 512 processors in an $8\times8\times8$ configuration. Suppose that $d_{\text{nonbonded}}=12$ Å, $r_{\text{bonded}}=4.5$ Å, $d_{\text{spreading}}=5$ Å, and $d_{\text{motion}}=0.1$ Å, all typical values for MD simulations of biomolecular systems. Using the midpoint method to parallelize the computation of the explicit nonbonded interactions would require an import volume of 7.9 nm^3. Assuming that the bonded force calculation is also parallelized using the midpoint method, no additional communication is required for bonded force computation, charge spreading, force interpolation, or particle position migration.

Using the HS method to parallelize the explicit nonbonded force computation would require an import volume of 14.0 nm^3. The import region of the HS method does not ensure that all particles within a distance $d_{\text{spreading}}$ of a particular box are imported onto that box, so each processor will need to import particles from an additional 2.9 nm^3 region to support charge spreading, for a combined import volume of 16.9 nm^3. Traditional parallelization strategies for bonded forces, in which each term is computed either on a fixed processor or on the processor containing a specific particle, will require additional communication, as will particle position migration.

Using the NT method to parallelize the explicit nonbonded force computation would require an import volume of 7.1 nm^3. Importing all particles within a distance $d_{\text{spreading}}$ of each interaction box, in order to support charge spreading, increases the combined import volume to 10.1 nm^3. One could reduce communication requirements slightly by applying the NT method to charge spreading (where the x and y base coordinates of the interaction box are those of the particle, while the z base coordinate is that of the mesh point). Some of the communication will then take the form of charges on mesh points, but if we assume that the amount of data associated with mesh points in a volume of space is comparable with the amount of data associated with particles in the same volume, then the amount of communication required is equivalent to that associated with a combined import volume of 9.8 nm^3. Optimization of the box aspect ratios can reduce the equivalent import volume to 9.2 nm^3, but this requires that one either adjust the aspect ratios of the global cell or use a grid of boxes with dimensions other than $8\times8\times8$. Even then, the equivalent import volume is greater than the 7.9 nm^3 required by the midpoint method, and the parallelization strategy based on the NT method, unlike one based on the midpoint method, will require additional communication for the computation of bonded forces and for particle position migration.

IV. MIDPOINT-ENSURED METHODS WITH IMPROVED COMPUTATIONAL LOAD BALANCE

We describe a particle as being *available* to a processor if the particle resides in the home box or import region of that processor. If multiple particles are all available to the same processor, we describe them as being *coavailable* to that processor. The import region of the midpoint method ensures that any set of particles that can be enclosed by a sphere of radius $R/2$ will be coavailable to at least one processor (the one whose home box contains their midpoint), but frequently, the particles will also be coavailable to one or

more additional processors. In Fig. 1, for example, particles 1 and 2 interact in the home box of particle 1 (the box labeled b), but these two particles are also coavailable in the boxes labeled a, d, and e. We can take advantage of the fact that sets of interacting particles are frequently coavailable on more than one processor to improve computational load balance among processors. We refer to a parallelization method whose import region includes that of the midpoint method as a midpoint-ensured method, regardless of how it assigns particle interactions to processors, because the fact that the box containing the midpoint of a set of interacting particles is guaranteed to house or import all of them ensures that at least one processor will be able to compute each required interaction.

In order to improve load balance, midpoint-ensured methods can assign interactions to processors based on the spatial distribution of particles. The more information processors share with one another about the particle distribution, the more effectively they can balance computational load. The relative performance of various midpoint-ensured methods depends on the relative costs of communication and computation, on the interaction radius and box size, and on prior statistical knowledge about the spatial distribution of particles. We leave the full exploration of the space of possible midpoint-ensured methods as an area for further work, but we present as an example one particular midpoint-ensured method that can significantly improve load balance with only minimal additional communication requirements.

For expository simplicity, we first describe this method for a one-dimensional space where each "box" is simply an interval of length b and where a set of m particles interact if they all fall within some sphere of radius $R/2$ (i.e., within some interval of length R). We also assume that R is no greater than b. We number the processors such that the home box of processor i is the ith box from the left. The computation proceeds in the following manner:

- Each processor imports all remote particles within a distance $R/2$ of its home box.
- Each processor finds its set of *locally computable interactions*, consisting of all combinations of m particles available to that processor that fall within a sphere of radius $R/2$. Each processor i divides its set of locally computable interactions into three nonoverlapping subsets: interactions that are also computable on its left neighbor (processor $i-1$), interactions that are also computable on its right neighbor (processor $i+1$), and interactions that are only computable on processor i. We denote these subsets L_i, R_i, and C_i, respectively, and we denote their sizes by l_i, r_i, and c_i. R_i and L_{i+1} contain exactly the same elements, namely, all combinations of m particles that are within a distance $R/2$ of the boundary between box i and box $i+1$, so $r_i=l_{i+1}$.
- Each processor i sends the value of c_i to its left and right neighbors. This is the only communication required by this parallelization method beyond that required by the midpoint method. Each processor i now has the values of c_{i-1} and c_{i+1} as well as c_i, l_i, and r_i.
- Each processor i determines which of the interactions in L_i and which of the interactions in R_i it will compute, and which of these interactions it will leave to its neighbors to compute. In order to ensure that each interaction is in fact computed somewhere, neighboring processors must make consistent decisions about the division of labor between them. Processors i and $i+1$ both have access to the values of c_i, c_{i+1}, and r_i (because $r_i=l_{i+1}$). Processor i will compute the first k_i interactions in R_i, while processor $i+1$ will compute the last r_i-k_i interactions, where processors i and $i+1$ independently compute identical values for k_i based on c_i, c_{i+1}, and r_i. To avoid missing or duplicating the computation of any interaction, processors i and $i+1$ must agree on the ordering of the equivalent sets R_i and L_{i+1}; such a consistent ordering might be based on the positions of the particles involved in each interaction, with ties broken by other particle characteristics such as charge. In the absence of prior statistical information about the spatial distribution of particles, each processor computes k_i as

$$k_i = \max\left(0, \min\left(r_i, \mathrm{round}\left(\frac{r_i}{2} + \frac{c_{i+1} - c_i}{3}\right)\right)\right). \tag{1}$$

  That is, if c_{i+1} and c_i are identical, then we assume that processors i and $i+1$ have a comparable load, and we therefore split the interactions in R_i evenly between them. If $c_{i+1} > c_i$, then we assign more than half the interactions in R_i to processor i, in an attempt to balance the load. The max, min, and round operations in Eq. (1) are necessitated by the fact that k_i can only take on integer values between 0 and r_i. In the absence of such restrictions, distributing interactions according to the formula $k_i=r_i/2+(c_{i+1}-c_i)/3$ for all i would result in the assignment of an identical number of interactions to processors i and $i+1$, if c_{i-1} and c_{i+2} were both equal to the average of c_i and c_{i+1}. This average represents a reasonable estimate of c_{i-1} and c_{i+2} based on information to which processors i and $i+1$ both have access. Better load balancing could be achieved if both processors had access to c_{i-1} and c_{i+2}, or if processors had some prior statistical information about the particle distribution (e.g., the average particle density).

Generalizations of this method to higher dimensions can take several forms. A computationally efficient and easily implemented generalization is available if we expand the import region slightly such that its outer boundary is a rectangular parallelepiped extending a distance $R/2$ beyond the interaction box in the positive and negative directions along each coordinate axis. We describe this generalization in the two-dimensional case, where the outer boundary of the import region will be a rectangle (a square if the interaction box is a square). Assuming that the side lengths of the interaction box are all larger than the import radius R, an individual interaction may be computable on four boxes that share a common corner, on two boxes that share a common edge, or only on a single box. Our strategy is first to assign each interaction that can be computed by two or four boxes in neighboring columns to a unique column, and then to assign

interactions that can be computed by boxes in neighboring rows to a unique row. We perform these assignments by repeated application of a procedure similar to that used to split R_i in the one-dimensional case, with appropriate values substituted for r_i, c_i, and c_{i+1} in Eq. (1).

More specifically, we assume that processor (i,j) is in the ith row and the jth column. After importing particles in its import region, processor (i,j) divides its set of locally computable interactions into nine subsets, based on which other processors can compute them. We denote these subsets by $S_{i,j}^{a,b}$, where a and b can each take on the values -1, 0, or 1. The value of a indicates whether the interactions in $S_{i,j}^{a,b}$ can be computed by processor $(i-1,j)$ (in which case $a=-1$), by processor $(i+1,j)$ (in which case $a=1$), or neither (in which case $a=0$). The value of b indicates whether the interactions can be computed by processor $(i,j-1)$ (in which case $b=-1$), by processor $(i,j+1)$ (in which case $b=1$), or neither (in which case $b=0$). The shape of the import region implies that an interaction is in $S_{i,j}^{1,1}$ if and only if it is computable in both processors (i,j) and $(i+1,j+1)$ (this would not be the case if the import volume were restricted to that of the midpoint method). Neighboring processors agree on their set of shared elements; for example, $S_{i,j}^{1,1}$ contains the same elements as $S_{i+1,j+1}^{-1,-1}$, $S_{i+1,j}^{-1,1}$, and $S_{i,j+1}^{1,-1}$. We denote the size of $S_{i,j}^{a,b}$ by $s_{i,j}^{a,b}$.

Each processor (i,j) sends the values $s_{i,j}^{1,0}$, $s_{i,j}^{0,0}$, and $s_{i,j}^{-1,0}$ to processors $(i,j-1)$ and $(i,j+1)$. For each a in $\{-1,0,1\}$, we assign the first $k_{i,j}^{a,1}$ interactions in $S_{i,j}^{a,1}$ to processors in column j and the last $s_{i,j}^{a,1}-k_{i,j}^{a,1}$ to processors in column $j+1$, where $k_{i,j}^{a,1}$ is computed by Eq. (1) with $s_{i,j}^{a,1}$ substituted for r_i, $s_{i,j}^{a,0}$ substituted for c_i, and $s_{i,j+1}^{a,0}$ substituted for c_{i+1}. Processor $(i,j+1)$ independently computes identical column assignments for all interactions in $S_{i,j}^{a,1}$. Processors $(i+1,j)$ and $(i+1,j+1)$ independently compute identical column assignments for interactions in $S_{i,j}^{1,1}$, and processors $(i-1,j)$ and $(i-1,j+1)$ independently compute identical column assignments for interactions in $S_{i,j}^{-1,1}$. At this point, all interactions have been uniquely assigned to a column of boxes. Interactions are then distributed between boxes within each column using a procedure identical to that described previously for the one-dimensional case.

Figure 6 shows the results of applying this algorithm to a three-dimensional system representing a protein solvated in explicit water. We determined particle positions by placing the reported crystal structure of the 213-residue protein Catechol O-Methyltransferase (PDB code 1vid)[26] in a water bath (neutralized with the appropriate number of counter ions), with interactions modeled using the OPLS-AA force field[19] and the SPC model for water.[27] A representative frame was generated by subjecting the solvated system to minimization followed by a 50 ps MD simulation at 300 K for equilibration. The entire chemical system contains 50 846 atoms in a cubic global cell measuring 80.0 Å per side. Figure 6(a) shows the number of atom pair interactions assigned to each processor using the midpoint method, assuming 64 processors in a $4\times4\times4$ configuration and an interaction radius of $R=12$ Å. Figure 6(b) shows the number of interactions assigned to each processor using the midpoint-ensured method we have described. The maximum number of interactions

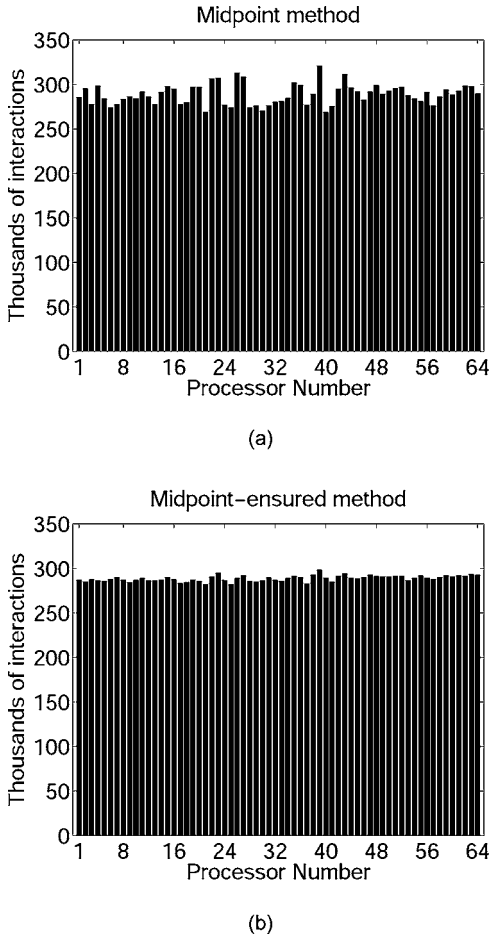


FIG. 6. Number of interactions assigned to each processor for a biomolecular system by (a) the midpoint method and (b) a midpoint-ensured method. The system contained 50 846 atoms in a cubic global cell measuring 80.0 Å per side, parallelized on 64 processors with $R=12$ Å.

assigned to a processor by the midpoint method exceeds the mean by 11.2%. With the midpoint-ensured method, the maximum exceeds the mean by only 3.4%. In this example, the import region of the midpoint-ensured method is larger than that of the midpoint method by 12.1%. One could formulate multidimensional midpoint-ensured methods that would require only the import volume of the midpoint method, at the cost of a more complex algorithm for assigning interactions to processors.

As we have noted previously,[8] neutral territory methods enjoy improved load balancing properties relative to traditional spatial decompositions when the number of processors is close to or larger than the number of particles in the system, because neutral territory methods can calculate interactions within processors that contain no particles. At high levels of parallelism, the midpoint method, the ES method, and the NT method will all result in better load balance than the HS method. At the level of parallelism in our example, however, the differences are negligible; the maximum number of interactions assigned to a processor exceeds the mean by between 9% and 13% for all four methods.

One can generalize the midpoint-ensured method described above in a straightforward manner to the case where R is greater than the smallest side length of the box by re-

quiring that an interaction between a set of particles always be computed either in the box containing their midpoint or in a neighboring box (i.e., a box that shares at least a corner with it). Other midpoint-ensured methods that remove this restriction can lead to further improvements in load balance at the cost of some additional communication or complexity. In some situations, generalizations beyond the scope of this paper may be required to achieve satisfactory load balance. For example, a midpoint-ensured method may take into account the fact that, in some simulations, certain interactions are more computationally expensive than others. Likewise, to compensate for coarse-scale variations in particle density, one might use a midpoint-ensured method where the regions assigned to different processors are of different shapes and sizes.

V. CONCLUSIONS

Unlike previously described neutral territory methods, the midpoint method applies to interactions involving more than two particles. It requires less communication bandwidth than traditional spatial decompositions whenever the global cell is partitioned into boxes along more than one dimension. The midpoint method also offers several other advantages over both traditional spatial decomposition methods and previously described neutral territory methods, including a more even distribution of bandwidth across communication links, a lower maximum hop count in communication networks with a toroidal mesh topology, and a smaller required number of send/receive cycles. The midpoint method proves applicable to problems in which each interaction involves multiple particles, with no required increase in import volume. Variants of the midpoint method known as midpoint-ensured methods allow for improved computational load balance among processors relative to both the midpoint method and traditional methods.

The midpoint method proves particularly effective for parallelization of MD simulations using common biomolecular force fields. It applies to the evaluation of both bonded force terms and explicit pairwise nonbonded force terms. Moreover, its data communication requirements are such that the communication that supports pairwise computation of nonbonded forces also suffices for bonded force computation, for particle position migration, and for the charge spreading and force interpolation computations associated with the efficient evaluation of long-range electrostatic interactions. At very high degrees of parallelism, certain other neutral territory methods require asymptotically lower communication bandwidth than the midpoint method.[8,9,14] In the course of implementing a parallel molecular dynamics package for a commodity LINUX cluster, however, we have been able to obtain better performance with the midpoint method than with either traditional methods or previously reported neutral territory methods for the parallelization of moderately sized systems (~50 000 atoms) up to at least 512 processing nodes; details of this package will be reported in a future paper.

ACKNOWLEDGMENTS

Thanks to Christine McLeavey for rendering many of the figures in this paper and to John Klepeis for preparing the chemical system used as an example to assess load balancing properties of different parallelization methods (Fig. 6). We have benefited from fruitful discussions with Mike Eastwood, Edmond Chow, and members of the GROMACS[3] development team, particularly Berk Hess, Erik Lindahl, and David van der Spoel.

APPENDIX: INTERACTIONS BETWEEN SETS OF THREE OR MORE PARTICLES

One disadvantage of the midpoint method for interactions that involve three or more particles is that determining the center of the smallest sphere containing m particles requires a nontrivial amount of computation when $m \geqslant 3$. One alternative is to use midpoint-ensured methods that do not require computation of a midpoint. Another is to use an efficient approximation for the midpoint. Two such approximations are:

(1) For each of the coordinate dimensions, approximate the coordinate of the midpoint as the arithmetic mean of the coordinates of the particles.
(2) For each of the coordinate dimensions, approximate the coordinate of the midpoint as the arithmetic mean of the maximum and the minimum of the coordinates of the particles.

When using either approximation, one needs to expand the import region slightly to ensure that the box containing the approximate midpoint of a set of interacting particles will import all remote particles in that set. In the absence of any restrictions on the spatial distribution of particles, use of approximation (1) requires that the import region contain all particles within a distance $R(1-1/m)$ of the interaction box boundary, implying a substantial increase in import volume (if $m-1$ particles are collocated inside the interaction box at a distance R/m from the boundary and the mth point is located outside the interaction box at a distance just under $R(1-1/m)$ from the boundary, then the midpoint will fall within the boundary). In MD simulations using common biomolecular force fields, however, a smaller increase in import volume may be sufficient, because the spatial arrangement of the particles involved in a bonded term is not arbitrary; for example, any pair of atoms must maintain some minimal distance from one another under normal conditions.

A sufficient import region for use of approximation (2) is that enclosed by a rectangular parallelepiped extending a distance $R/2$ beyond the interaction box in the positive and negative directions along each coordinate axis. One can show that a somewhat smaller region suffices, one that includes all remote particles within a rectangular parallelepiped extending a distance $R/4$ beyond the interaction box in the positive and negative directions along each coordinate axis as well as all particles within a distance $R/4$ of that parallelepiped. In three or fewer dimensions, for $m \geqslant 3$, this

import volume is always smaller than that required by approximation (1) in the absence of restrictions on the spatial distribution of particles.

The generalization of the ES method to interactions involving three or more particles (Sec. II) also requires expanding the import region, because a set of particles that interact in a particular box under this method may include a particle at a distance greater than R from the box. A sufficient import region includes all remote particles within a rectangular parallelepiped extending a distance $R/2$ beyond the interaction box in the $+x$, $+y$, and $+z$ directions, as well as all particles within a distance $R/2$ of that parallelepiped whose x, y, and z coordinates are larger than the base coordinates of the interaction box. This import region has the same volume as the smaller of the aforementioned sufficient import regions for use of approximation (2) with the midpoint method; this volume is larger than the import volume of the exact midpoint method.

[1] M. Karplus and J. A. McCammon, Nat. Struct. Biol. **9**, 646 (2002); C. L. Brooks and D. A. Case, Chem. Rev. (Washington, D.C.) **93**, 2487 (1993); D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed. (Academic, London, 2001); W. Wang, O. Donini, C. M. Reyes, and P. A. Kollman, Annu. Rev. Biophys. Biomol. Struct. **30**, 211 (2001); T. Schlick, R. D. Skeel, A. T. Brunger, L. V. Kalé, J. A. Board, J. Hermans, and K. Schulten, J. Comput. Phys. **151**, 9 (1999).
[2] G. S. Almasi, C. Cascaval, J. G. Castanos *et al.*, Int. J. Parallel Prog. **30**, 317 (2002).
[3] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, J. Comput. Chem. **26**, 1701 (2005).
[4] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Adam Hilger, Bristol, 1988).
[5] J. J. Monaghan, Annu. Rev. Astron. Astrophys. **30**, 543 (1992).
[6] G. S. Heffelfinger, Comput. Phys. Commun. **128**, 219 (2000); S. Plimpton, J. Comput. Phys. **117**, 1 (1995).
[7] K. J. Bowers, R. O. Dror, and D. E. Shaw, J. Phys.: Conf. Ser. **16**, 300 (2005).
[8] K. J. Bowers, R. O. Dror, and D. E. Shaw, J. Comput. Phys. (in press).
[9] D. E. Shaw, J. Comput. Chem. **26**, 1318 (2005).
[10] B. G. Fitch, A. Rayshubskiy, M. Eleftheriou *et al.*, *Blue Matter: Strong Scaling of Molecular Dynamics on Blue Gene/L*, IBM Report RC23688, IBM (2005).
[11] B. G. Fitch, A. Rayshubskiy, M. Eleftheriou *et al.*, *Blue Matter: Strong Scaling of Molecular Dynamics on Blue Gene/L*, IBM Report RC23888, IBM (2006).
[12] One can only guarantee that an interaction between two particles residing in the same processor will be computed on that processor if the region assigned to that processor is convex.
[13] We have discovered novel neutral territory methods whose import volume is slightly smaller than that of the midpoint and ES methods for small α_R. These methods will be described in a subsequent paper.
[14] M. Snir, Theory Comput. Syst. **37**, 295 (2004).
[15] These figures are approximate because we have ignored the constraints on box aspect ratios due to the finite number of processors. In order for the boxes to be exactly cubic when the global cell is cubic, for example, the number of processors must be the cube of some integer. In practice, one might choose not to use a few of the available processors in order to obtain more convenient aspect ratios.
[16] *Red Storm System Raises Bar on Supercomputer Scalability* (Cray, Seattle, 2003); N. R. Adiga, G. Almasi, G. S. Almasi *et al.*, in *Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, Jersey City, NJ, 207 (2005); R. E. Kessler and J. L. Schwarzmeier, in *38th IEEE Comput. Soc. Intl. Conf.*, 176 (1993); S. Scott and G. Thomas, in *Proceedings of Hot Interconnects IV*, 147 (1996).
[17] J. MacKerell, D. Bashford, M. Bellott *et al.*, J. Phys. Chem. B **102**, 3586 (1998).
[18] P. A. Kollman, R. W. Dixon, W. D. Cornell, T. Fox, C. Chipot, and A. Pohorille, in *Computer Simulations of Biological Systems*, edited by W. F. van Gunsteren (Kluwer, Dordrecht, Netherlands ESCOM, Leiden, 1997), Vol. 3, 83.
[19] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).
[20] W. R. P. Scott, P. H. Hünenberger, I. G. Tironi, A. E. Marks, S. R. Billeter, J. Fennen, A. E. Torda, T. Huber, P. Krüger, and W. F. van Gunsteren, J. Phys. Chem. A **103**, 3596 (1999).
[21] T. A. Halgren, J. Comput. Chem. **20**, 730 (1999).
[22] C. L. Brooks, B. M. Pettit, and M. Karplus, J. Chem. Phys. **83**, 5897 (1985); P. Mark and L. Nilsson, J. Comput. Chem. **23**, 1211 (2002); J. Norberg and L. Nilsson, Biophys. J. **79**, 1537 (2000); M. Patra, M. Karttunen, T. Hyvönen, E. Falck, P. Lindqvist, and I. Vattulainen, Biophys. J. **84**, 3636 (2003).
[23] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993); U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[24] C. Sagui and T. Darden, J. Chem. Phys. **114**, 6578 (2001).
[25] Y. Shan, J. L. Klepeis, M. P. Eastwood, R. O. Dror, and D. E. Shaw, J. Chem. Phys. **122**, 054101 (2005).
[26] J. Vidgren, L. A. Svensson, and A. Liljas, Nature (London) **368**, 354 (1994).
[27] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans, in *Intermolecular Forces*, edited by B. Pullman (Reidel, Dordrecht, 1981), 331.