Extending Parallel Scalability of LAMMPS and Multiscale Reactive Molecular Simulations

Yuxing Peng[1]
University of Chicago
yuxing@uchicago.edu

Chris Knight[2]
Argonne National Laboratory
knight@mcs.anl.gov

Philip Blood
Pittsburgh Supercomputing Center
Carnegie Mellon University
blood@psc.edu

Lonnie Crosby
National Institute for Computational Sciences
University of Tennessee, Knoxville
lcrosby1@utk.edu

Gregory A. Voth[1,2]
University of Chicago and Argonne National Laboratory
gavoth@uchicago.edu

ABSTRACT

Conducting molecular dynamics (MD) simulations involving chemical reactions in *large-scale* condensed phase systems (liquids, proteins, fuel cells, etc…) is a computationally prohibitive task even though many new *ab initio* based methodologies (i.e., AIMD, QM/MM) have been developed. Chemical processes occur over a range of length scales and are coupled to slow (long time scale) system motions, which make adequate sampling a challenge. Multistate methodologies, such as the multistate empirical valence bond (MS-EVB) method, which are based on effective force fields, are more computationally efficient and enable the simulation of chemical reactions over the necessary time and length scales to properly converge statistical properties.

The typical parallel scaling bottleneck in both reactive and nonreactive all-atom MD simulations is the accurate treatment of long-range electrostatic interactions. Currently, Ewald-type algorithms rely on three-dimensional Fast Fourier Transform (3D-FFT) calculations. The parallel scaling of these 3D-FFT calculations can be severely degraded at higher processor counts due to necessary MPI all-to-all communication. This poses an even bigger problem in MS-EVB calculations, since the electrostatics, and hence the 3D-FFT, must be evaluated many times during a single time step.

Due to the limited scaling of the 3D-FFT in MD simulations, the traditional single-program-multiple-data (SPMD) parallelism model is only able to utilize several hundred CPU cores, even for very large systems. However, with a proper implementation of a multi-program (MP) model, large systems can scale to thousands of CPU cores. This paper will discuss recent efforts in collaboration with XSEDE advanced support to implement the MS-EVB model in the scalable LAMMPS MD code, and to further improve parallel scaling by implementing MP parallelization algorithms in LAMMPS. These algorithms improve parallel scaling in both the standard LAMMPS code and LAMMPS with MS-EVB, thus facilitating the efficient simulation of large-scale condensed phase systems, which include the ability to model chemical reactions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
XSEDE12, July 16-19, 2012, Chicago, IL, USA.
Copyright 2012 ACM 1-58113-000-0/00/0010…$10.00.

Categories and Subject Descriptors

J.2 **[Physical Sciences and Engineering]**: Chemistry

General Terms

Algorithms, Performance, and Theory

Keywords

Reactive Molecular Dynamics, MS-EVB, LAMMPS

1. INTRODUCTION

The study of chemical reactions is challenging, but is required to model a number of important problems in chemistry and biology, such as the coupling between proton shuttling and large-scale characteristics of the environment. Recent advances in *ab initio* simulation methodologies have greatly facilitated the study of chemical reactions in condensed phase by explicitly treating the electronic degrees of freedom, which naturally account for the formation and breaking of chemical bonds. The computational cost of these methods, however, is large and limits the applicability to relatively small systems and short timescales. In order to statistically converge physical properties across multiple length and time scales, such as those necessary to accurately model the transport of an excess proton or hydroxide ion in aqueous solutions, or even observe the phenomena of interest, such as large-scale protein motion, one would need to resort to less accurate *ab initio* methods. One promising approach to circumvent these difficulties is the use of reactive multi-configurational methods that describe the system as a linear combination of several bonding topologies. The Multistate Empirical Valence Bond (MS-EVB) method is one example in this general class of methods that has been developed in the Voth group over the past fifteen years to address the challenges of modeling chemical reactions in condensed phase environments.[1,

2] When properly parameterized against high-level electronic structure methods, the MS-EVB approach can be viewed as an extension of the accurate *ab initio* simulation methods, but at a significantly reduced computational cost.

Significant progress in understanding the solvation structure and dynamics of the excess proton for a range of condensed phase systems has been achieved over the years using resources provided by the TeraGrid/XSEDE program. With the first generation MS-EVB code, DL_EVB (see Section 3), a number of important problems have been investigated [1] and only a small selection of recent results will be highlighted here. Recent work on cytochrome c oxidase, a redox-driven proton pump, which plays several important roles in aerobic cellular respiration, has found the importance of the role played by the second subunit and a redox dependence on the electrochemical proton gradient as well as the sensitivity of results to a key mutation.[3] With the analysis of reactive simulations, there have been new insights in understanding the proton permeation in aquaporin water channels that have identified those residues important for impeding proton permeation, which were experimentally confirmed.[4] A new force-matching algorithm was developed that enables the efficient parameterization of reactive MS-EVB models by directly using the forces calculated from a condensed phase *ab initio* simulation.[5]

Although a number of scientific breakthroughs have been made possible with the original DL_EVB code, there were a number of challenges that needed to be surmounted if the MS-EVB methodology was going to remain a viable solution to reactive molecular dynamics. In the following sections, a brief overview of the MS-EVB algorithm is presented, which is then followed by discussion on the challenges that needed to be tackled in order to take full advantage of the generality of the MS-EVB algorithm as well as significantly improve the parallel performance of the simulations. The focus will then turn to the new and improved MS-EVB implementation within the LAMMPS MD code,[6] in collaboration with TeraGrid/XSEDE advanced support, and highlighting some of the recent scientific breakthroughs that have already been achieved using XSEDE resources. This is then followed by a short discussion on recent scalability issues, which have largely been addressed with the implementation of a Multiple Program parallelization strategy for calculating long-ranged electrostatic interactions. This paper concludes with a short discussion of performance issues under current investigation and a future outlook.

2. MS-EVB

Typical molecular mechanics force fields assume a fixed bonding topology through the course of a simulation, which makes the modeling of chemical reactions impossible. Multistate algorithms, such as MS-EVB, describe the system not as a single, fixed bonding topology, but as a linear combination of several possible bonding topologies. The MS-EVB algorithm is a generalization of the original EVB algorithm,[7] whereby the possible bonding topologies to be considered are determined "on the fly" over the course of a simulation. As an example, for the case of the hydrated proton in which transport occurs via the Grotthuss mechanism, this selection process would start by considering all water molecules in the vicinity (Fig. 1). These water molecules (newly formed hydronium cations) would then be considered to donate a proton to water molecules proximal to them and so on until a sufficient number of topologies have been identified, which for the aqueous proton involves 3-4 proton hops to capture all significant contributions.

The collection of atoms and molecules that are considered for possible reactions form the reactive complex and the collection of bonding topologies is taken to form the basis of a quantum-like Hamiltonian matrix, where the diagonal entries correspond to unique bonding topologies (diabatic states) and the off-diagonal entries couple two states and serve as the mechanism for enabling transitions between states. The Hamiltonian matrix is then diagonalized and the coefficients of the ground state eigenvector are used in combination with the Hellman-Feynman theorem to calculate atomic forces. These forces are then use as input to a molecular dynamics code to integrate Newton's the equations of motion and propagate particles in time.

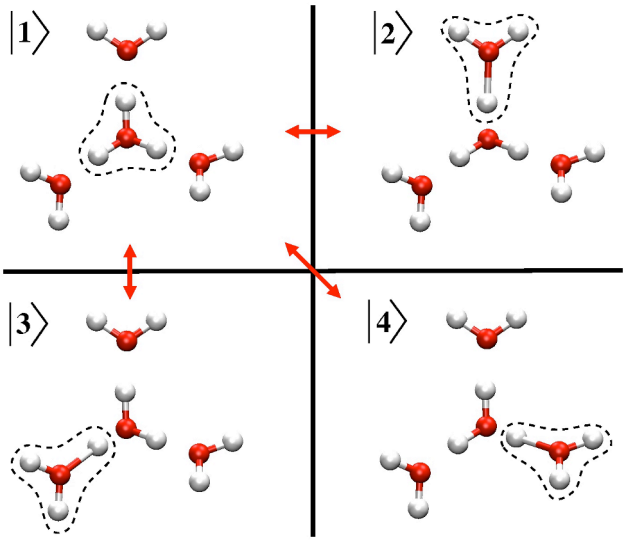


Figure 1. Illustrative example of bonding topologies considered for the hydrated excess proton (Figure 2 of Ref. [2]). Arrows indicated an off-diagonal coupling between diabatic states.

3. EARLY CHALLENGES: DL_EVB

One of the main challenges to efficiently calculating a MS-EVB simulation is that all of the nonzero matrix elements of the (sparse) Hamiltonian matrix need to be evaluated at every step in the simulation. Initial breakthroughs in resolving this issue were to logically separate the system into two subsystems: those atoms that were described by a variable bonding topology (within reactive complex) and those atoms where the topology remained fixed at a given MD step (outside reactive complex). With this separation, the interactions involving those atoms in the environment region needed to be evaluated only once per MD step. This leads to a significant boost in computational performance particularly when calculating the long-range k-space portion of the electrostatic interactions.[8] However, even with this strategy for the efficient calculation of energies and forces, the original DL_EVB implementation (within the DL_POLY2.0 MD code [9]) faced serious scaling issues across even a few nodes as well as a strict upper limit on the system size that could be studied due to memory management. There were also restrictions based on the original implementation that presented difficulties in exploiting the full generality of the MS-EVB algorithm to chemical reactions beyond the exchange of a single proton between two molecules. With the required length and time scales of important scientific problems requiring a fast and highly

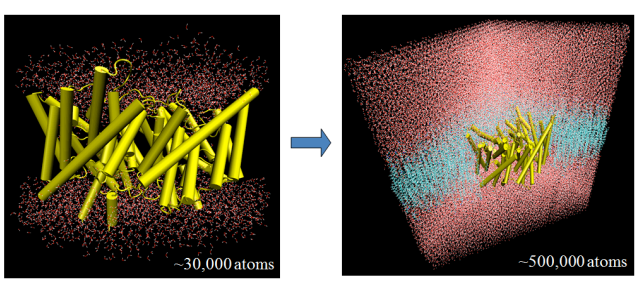
scalable code capable of spanning thousands of processors, significant effort by the Voth group has been focused on the planning and implementation of such a code. In collaboration with Teragrid/XSEDE advanced user support over the past several years, a more robust and scalable code has been developed.

4. RAPTOR CODE

In recent years, the frequency of a single microprocessor has plateaued at around 4.0 GHz, although modern supercomputers employ multi-core processors with lower operating frequencies, together with high-speed networks, to maximize peak performance while reducing power consumption. With these new high-performance computing architectures, the design of highly scalable programs capable of spanning thousands of processors is of paramount importance. As noted previously, the original MS-EVB code, DL_EVB, facilitated many important insights into complicated multiscale processes involving proton transport, but was limited due to poor scalability. To facilitate deeper insight into even more challenging systems, a significantly improved version of the MS-EVB implementation was required. In collaboration with TeraGrid/XSEDE advanced support, this has largely been accomplished in the Voth group's recently developed RAPTOR (Rapid Approach for Proton Transport and Other Reactions) code, which is written as an add-on package for the LAMMPS MD code.[6]

The LAMMPS code is highly scalable, modular, and has proven straightforward in extending and maintaining additional features. The MS-EVB algorithm is written as a "fancy" potential, which is interfaced to LAMMPS through the "fix" mechanism. The modular structure and philosophy of LAMMPS is continued in RAPTOR, thus allowing a straightforward implementation of new methods and algorithms with little to no modification of the original LAMMPS source.

With the extensible framework provided by the LAMMPS (and hence RAPTOR) code, the number and types of problems that can be studied has been significantly extended compared to the DL_EVB code. This strategy for the RAPTOR code has already proven invaluable by enabling simulations of chemical reactions beyond proton transport. The current implementation of the MS-EVB methodology in RAPTOR is sufficiently flexible to enable the modeling of generic chemical reactions, such as those involved in ATP hydrolysis, which include several decomposition and synthesis reactions in addition to proton transfer with water. RAPTOR can also use the SCI-MS-EVB algorithm to run large-scale simulations of concentrated systems containing many reactive protons and hydroxide ions.



No lipid & restrained backbone — All atom membrane and protein

Figure 2. RAPTOR significantly improves the simulation capability of the MS-EVB method.

The superior parallel decomposition strategy in LAMMPS (spatial decomposition) over DL_POLY-based DL_EVB resulted in immediate performance gains in RAPTOR. When simulating a 40K atom system in DL_EVB, it proved difficult to scale beyond a single node (16 cores) on Ranger, yielding only ~25 picoseconds of simulation time per day. The original RAPTOR code was nearly 10x faster, yielding 200 picoseconds/day for a similar system on 1 node of Ranger. The ability to simulate larger systems for much longer time scales has significantly increased the science problems that can be addressed using the MS-EVB methodology. For example, in the DL_EVB code, the simulation of a membrane protein was only possible when the membrane environment was modeled using a partially frozen protein with fixed backbone atoms. This procedure likely introduces artifacts into the simulations by not capturing the coupling of long time motions of the membrane environment to the proton transport process. With the RAPTOR code, the simulation of this type of system can be done with higher fidelity, using an all-atom representation of the membrane and solvent (Fig. 2).

5. PARALLEL SCALING ISSUES

Although the initial RAPTOR code based on LAMMPS had improved parallel scaling compared to the DL_EVB code, the scalability of RAPTOR was still extremely limited beyond tens of cores. The calculation of long-ranged electrostatics for each nonzero matrix element of the MS-EVB Hamiltonian is a serious bottleneck, since electrostatic interactions between atoms contributing to each matrix element and the rest of the system must be calculated at every time step. Long-ranged electrostatic interactions can be efficiently calculated using Ewald-type mesh algorithms, such as PPPM,[10] but these algorithms rely on 3D-FFT operations, which can degrade parallel efficiency at high processor counts. For each PPPM call, 4 3D-FFTs must be performed (1 forward, 3 reverse), each of which requires all-to-all communication. In RAPTOR, this means that 4 3D-FFTs are required for each diagonal element and 9 3D-FFTs for the interaction energies in the off-diagonal coupling. The exact number of 3D-FFT calls that must be made varies depending on the number of possible bonding topologies there are in the system at any given time step, but at a minimum this will happen many tens of times per time step. When one considers that a regular, nonreactive MD code, which is already difficult to scale, only performs a single PPPM (or equivalent) calculation per time step, it becomes apparent that getting the RAPTOR code to scale, without resorting to approximations, is an *extremely* challenging prospect.

A parallel speed-up analysis of LAMMPS has been done on Ranger and the results are shown in Fig. 3. The real space calculation shows an ideal parallel scaling with increasing number of nodes, however, the k-space portion of the calculation shows very poor speed-up with the performance decreasing with number of nodes. Further analysis confirmed that the main cause for the decreased performance in the k-space portion of the calculations is largely due to the communication intensive FFT operations.

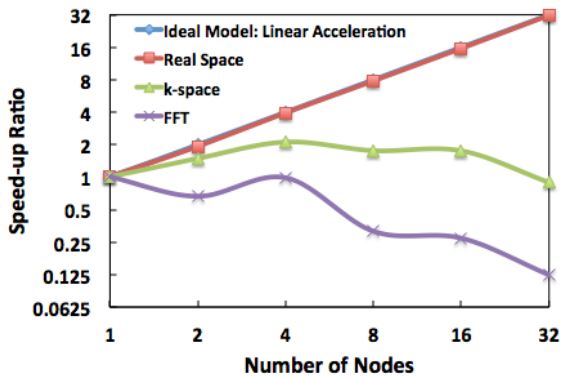


Figure 3. Parallel speed-up analysis on LAMMPS for its different calculation tasks. Axes are on a logarithmic scale. The simulations were done on Ranger, which has 16 cores per node.

6. MULTIPLE PROGRAM APPROACH

As stated above, the bottleneck of improving the scalability of the simulations relies on improving the performance of the all-to-all communications in FFT operations. A promising solution to minimizing the costs of the FFT operations for simulations on large number of processors is a Multiple Program (MP) approach, similar to that implemented in GROMACS.[11] A traditional parallel MD simulator (i.e., LAMMPS) usually uses the Single-Program-Multiple-Data (SPMD) model, in which all processors are running the same instructions, but operating on different sets of data. In the MP approach (Fig. 4), a separate partition of processors is organized to execute the k-space specific operations (referred as "k-space partition"), while the remaining set of processors (referred as "r-space partition") are responsible for all remaining parts of the calculation, such as maintaining the neighbor list, evaluating short-range interactions, and integrating Newton's equations of motion. An optimal choice for the relative size of the partitions for the r- and k-space partitions was determined to be 3:1, which is in agreement with that used in other MD codes.[11] Therefore, only one fourth of the processors are involved in FFT, which leads to a large reduction in the cost of the all-to-all communication.

As already stated, similar solutions have been applied in other popular MD simulation programs, such as GROMACS and NAMD, that improve the computational scalability of these programs up to more than 1,000 processors. With this implementation of a MP parallelization strategy in LAMMPS, the scalability of condensed phases simulations with long-range interactions has been extended to higher processor counts. The implementation of this MP strategy was relatively straightforward thanks to the well-organized framework of the LAMMPS code, In particular, the MP approach was implemented as an add-on package (referred as "verlet/split" since Dec. 11, 2012) and facilitated by the "-partition" command, which was already present in LAMMPS and was used to create a two-partition task in parallel. The MP approach was similarly implemented for use in reactive simulations with the RAPTOR code..

In the MP model the processors in the two partitions execute different instructions, but share the same data blocks. Therefore, there is some degree of inter-partition communication required at each step in the simulation. However, the extra cost associated with this inter-partition communication is minor compared to the significant benefits from reducing the all-to-all communication of the FFT operations. First, this is due to the fact that the inter-partition communication is a local type of data-transfer, for example, it occurs between sets of four processors in the case of 3:1 partition ratio. Second, the amount of inter-partition communication is proportional to $1/P$ (P is the number of processors), while the all-to-all communication in FFTs is proportional to $\log P$.

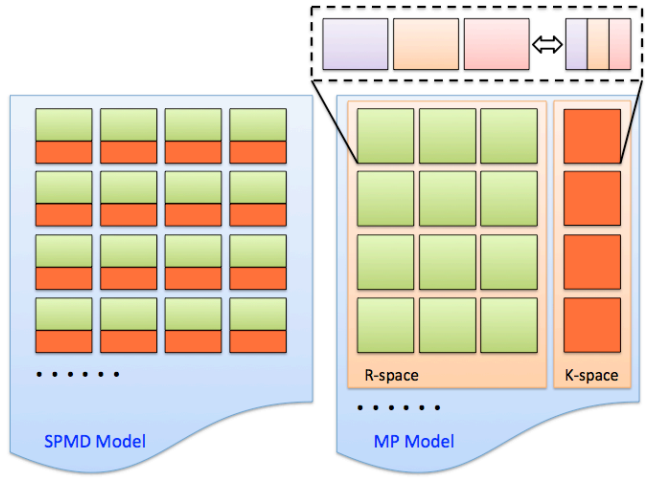


Figure 4. SPMD model *vs.* MP model. In two panels, the red blocks represent the computational tasks for k-space, which requires the all-to-all communications in FFT operations. The green ones represent the r-space tasks. The dashed frame shows the inter-partition data sharing in each row of the processors, and different colors represent different blocks of data.

6.1 Rank Reordering

Intra-socket and intra-node communication in multicore platforms is much faster than inter-node communication through the network. If one does not consider the varying computer hardware levels when designing the MPI communication scheme, then the overall performance will depend on the slowest network transfer. Different data-transfer topologies will have different efficiencies, which is one reason why hybrid programming models that combine OpenMP and MPI have recently become popular. To further improve the parallel efficiency of the MP approach presented here for both LAMMPS and RAPTOR, one can design a strategy so that the inter-partition data transfer occurs only at the inter-core level by reassigning the MPI ranks for all processes.

Two different plans for assigning MPI ranks and the corresponding inter-partition data transfer is shown in Fig. 5. Normally, MPI ranks the computing cores sequentially, therefore, the inter-partition data transfer occurs in both intra-node and inter-node levels (plan 1). Using a different assignment of MPI ranks to computing cores, plan 2 can be implemented, which only requires intra-node data transfers and should be more efficient than plan 1.

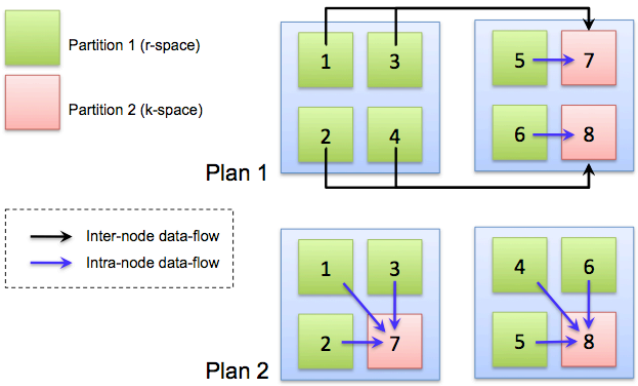


Figure 5. Different mapping plans between MPI ranks (black numbers) and CPU (blue blocks) cores, the resulting partitions (green and red blocks) and the data-flow (arrows) during inter-partition MPI communications.

6.2 Multiple Program LAMMPS Performance

All the benchmarks presented in this paper have been calculated using an all-atom Cytochrome c Oxidase (C*c*O) system (Fig. 6). C*c*O is a multi-subunit transmembrane protein that catalyzes the aerobic respiration in cells. The target model contains ~159k atoms with full protein subunits, explicit bilayer membrane and water solvation. All simulations have been conducted using the short-range cut-off of 12.0 Å and PPPM with precision 1E-4 (single-precision FFT).

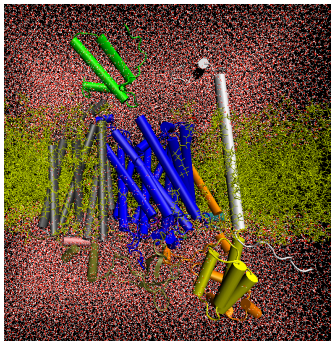


Figure 6. The snapshot of the benchmark system C*c*O, with all-atom models of proteins (sticks), cellular membrane (yellow) and water solvation box (red).

The performance of the new multiple program (MP) LAMMPS code was tested on various XSEDE platforms, including Kraken (NICS) and Ranger (TACC). The scaling performance for the LAMMPS code is shown in Figure 7. As discussed earlier, a key factor in the performance of the MP code is determining the optimal ratio of processors assigned to the real space and k-space calculations. In most cases, a real space to k-space ratio of 3:1 provides the best overall performance. At lower processor counts (<128 cores), the extra effort required to load balance the real space and k-space partitions outweighs the benefit of reducing the amount of communication arising from the k-space calculation, so the traditional single program approach performs equivalently or slightly better than the MP approach. The benefits of the MP parallelization strategy begin to appear at larger processor counts, where the communication intensive FFT operations start to dominate the total runtimes.

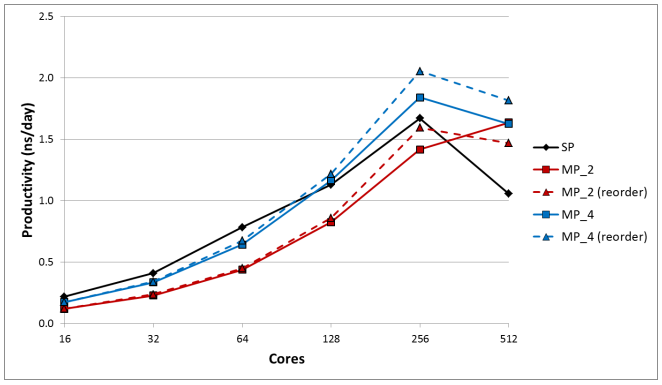


Figure 7. Improved scalability of MP LAMMPS running the 159 K atom CcO benchmark (nonreactive MD). SP: Original (Single Program) LAMMPS code; MP_2: Multiple Program LAMMPS with a 1:1 ratio of real space and k-space cores; MP_4: Multiple Program LAMMPS with a 3:1 ratio of real space and k-space cores. Multiple Program benchmarks were run with and without rank reordering as indicated.

In addition to finding the right r-space/k-space ratio, it was important to reorder the MPI ranks so that the real space processors and k-space processors responsible for the same simulation volume were placed close to each other, such as on the same node as illustrated in Figure 5. Initially, rank reordering for the LAMMPS benchmarks was tested on Kraken by setting the MPICH_RANK_REORDER_METHOD environment variable to specify a custom rank placement file. This type of manipulation is no longer necessary as the ability to control rank placement was recently added to the main LAMMPS distribution to support the new multiple program separation of r-space and k-space calculations. By choosing the optimal real and k-space ratio and optimizing rank placement, the peak scaling of MP LAMMPS improves by 23% in the benchmark in Figure 7. These enhancements, done in collaboration with XSEDE Extended Collaborative Support staff, are available to all LAMMPS users who require 3D-FFTs as part of their calculation.

6.3 RAPTOR Performance

The MP parallelization strategy was next implemented in the RAPTOR code to improve the parallel scaling of simulations for chemical reactions. In this case, the k-space contribution to the long-ranged electrostatic interactions for every nonzero element of the MS-EVB Hamiltonian matrix was evaluated on the smaller partition of processors. As seen in Figures 8 and 9 for Ranger (TACC) and Kraken (NICS), respectively, the parallel scaling of the RAPTOR code was significantly extended. The test system for these benchmarks is identical to that used for the nonreactive LAMMPS benchmarks discussed in the previous section. It is again found that the optimal ratio of real space to k-space processors is 3:1 and that a remapping of the MPI ranks was critical to improved performance, particularly above processor counts of one hundred, although an enhancement in productivity is generally observed independent of the number of processors for the RAPTOR simulations, which involve a large number of FFT operations at each simulation step.

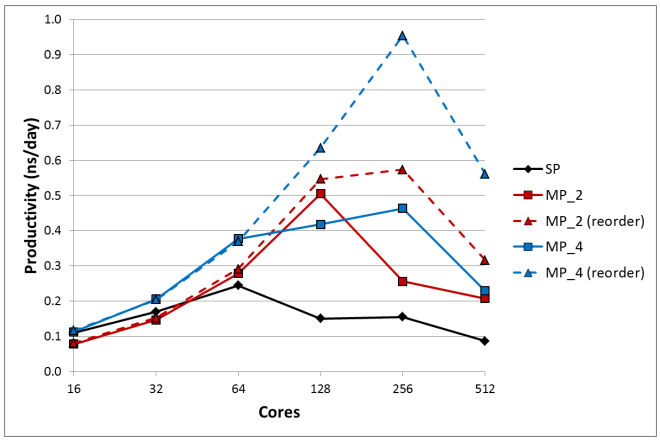


Figure 8. RAPTOR performance running reactive CcO simulations on the Ranger Sun Constellation Cluster at TACC.

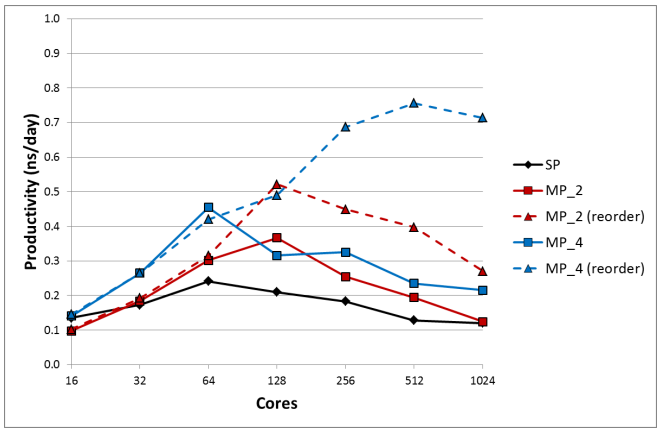


Figure 9. RAPTOR performance running reactive CcO simulations on the Kraken Cray XT5 at NICS.

6.4 Scaling Further

In a multiple program RAPTOR run, when the k-space partition reaches a core count at which the single program approach stops scaling (e.g. 64 cores for the CcO benchmark), the multiple program code also stops scaling. In the CcO benchmarks in Figures 8 and 9, when using a 3:1 r-space to k-space partition, this point is reached when the total MP RAPTOR simulation is using 256 cores. At 512 cores, the k-space partition is using 128 cores, which is where scaling significantly drops in a single program (SP) run. This may be offset somewhat by running an even smaller k-space partition at higher core counts, but this in itself will not yield good scalability due to the small amount of time spent in real space computation at very high core counts, and the need to balance this with the amount of k-space computation.

Indeed, the RAPTOR code is already doing exceedingly well, considering it is doing many tens of 3D-FFTs every time step, falling behind the regular nonreactive MD simulation in Figure 7 by only a factor of 2. Overall, the latest MP RAPTOR code provides a ~20-30x performance improvement over the DL_EVB code.

Despite the excellent results obtained with RAPTOR so far, there is further opportunity for improvement. Currently, calculations over the various EVB states are done one at a time, even though these calculations are independent of each other. One strategy currently being considered is to divide the simulation into many partitions (i.e. more than two), and perform the calculations over various EVB states simultaneously. This would allow the FFTs associated with those EVB states to run on smaller partitions, even for large simulations, thus saving on communication cost. In addition, running many independent calculations would provide an opportunity to overlap computation with communication, and hide latency. These opportunities will be explored through continued collaboration with XSEDE staff.

7. FUTURE OUTLOOK

The work discussed in this paper serves as the foundation for the implementation of additional parallelization strategies to further extend the parallel scalability of reactive molecular simulations. With the multiple program approach discussed here implemented in the main LAMMPS MD code, the general LAMMPS community and XSEDE users will also be able to take advantage of improved performance when long-range electrostatic interactions are required in their simulations. These algorithmic improvements will facilitate a broad range of science within the fields of chemistry, biology, and materials science, allowing researchers to explore chemical phenomena not previously accessible. With a continued collaborative effort with XSEDE advanced support, the parallel scaling of RAPTOR (and LAMMPS) will continue to improve, in particular for the simulation of concentrated systems containing many reactive protons and hydroxide ions, and take advantage of state-of-the-art computing resources that enable the efficient simulation of chemical reactions in large-scale condensed phase systems.

8. ACKNOWLEDGMENTS

Y.P. acknowledges support by the National Institutes of Health (NIH grant R01-GM053148). C.K. acknowledges support by the U.S. Department of Energy under contract DE-AC02-06CH11357 and an Argonne Computational Postdoctoral Fellowship. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Thanks to Steve Plimpton and LAMMPS development team for incorporating the MP approach into the main release of the LAMMPS code.

9. AUTHOR AFFILIATIONS

1) Department of Chemistry, James Franck Institute, Institute for Biophysical Dynamics, and Computation Institute, University of Chicago, Chicago, IL 60637

2) Computing, Environment, and Life Sciences, Argonne National Laboratory, Argonne, IL 60439

10. REFERENCES

[1] Swanson, J.M.J., Maupin, C.M., Chen, H., Petersen, M.K., Xu, J., Wu, Y., and Voth, G.A., 2007. Proton Solvation and Transport in Aqueous and Biomolecular Systems: Insights from Computer Simulations. *J. Phys. Chem. B 111*, 17, 4300-4314.

[2] Knight, C. and Voth, G.A., 2012. Coarse-graining away electronic structure: a rigorous route to accurate condensed phase interaction potentials. *Mol. Phys.*, DOI: 10.1080/00268976.2012.66821.

[3] Yamashita, T. and Voth, G.A., 2012. Insights into the Mechanism of Proton Transport in Cytochrome c Oxidase. *J. Am. Chem. Soc. 134*, 2, 1147-1152.

[4] Li, H., Chen, H., Steinbronn, C., Wu, B., Beitz, E., Zeuthen, T., and Voth, G.A., 2011. Enhancement of proton

conductance by mutations of the selectivity filter of aquaporin-1. *J. Mol. Biol. 407*, 4, 607-620.

[5] Knight, C., Maupin, C.M., Izvekov, S., and Voth, G.A., 2010. Defining Condensed Phase Reactive Force Fields from ab Initio Molecular Dynamics Simulations: The Case of the Hydrated Excess Proton. *J. Chem. Theory Comput. 6*, 10, 3223-3232.

[6] Plimpton, S.J., 1995. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comp. Phys. 117*, 1-19.

[7] Warshel, A., 1991. *Computer Modeling of Chemical Reactions in Enzymes and Solutions*. John Wiley and Sons, New York, NY, USA.

[8] Smondyrev, A.M. and Voth, G.A., 2002. Molecular Dynamics Simulation of Proton Transport Near the Surface of a Phospholipid Membrane. *Biophys. J. 82*, 3, 1460-1468.

[9] Smith, W. and Forester, T.R., 1996. DL_POLY_2.0: A general-purpose parallel molecular dynamics simulation package. *Journal of Molecular Graphics 14*, 3, 136-141.

[10] Hockney, R.W. and Eastwood, J.W., 1988. *Computer Simulation Using Particles*. Taylor & Francis Group, New York.

[11] Hess, B., Kutzner, C., Spoel, D.V.D., and Lindahl, E., 2008. GROMACS 4: Algorithms for Highly Efficient Load-Balanced, and Scalable Molecular Simulation. *JCTC 4*, 3, 435-447.

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.