JOURNAL OF CHEMICAL INFORMATION AND MODELING

Application Note

Cite This: J. Chem. Inf. Model. 2018, 58, 2043−2050

pubs.acs.org/jcim

GPU-Accelerated Molecular Dynamics and Free Energy Methods in Amber18: Performance Enhancements and New Features

Tai-Sung Lee,[†] David S. Cerutti,[†] Dan Mermelstein,[‡] Charles Lin,[‡] Scott LeGrand,[§] Timothy J. Giese,[†] Adrian Roitberg,[∥] David A. Case,[†] Ross C. Walker,[*,⊥] and Darrin M. York[*,†]

[†]Laboratory for Biomolecular Simulation Research, Center for Integrative Proteomics Research and Department of Chemistry and Chemical Biology, Rutgers University, Piscataway, New Jersey 08854, United States

[‡]Department of Chemistry and Biochemistry, University of California, San Diego, La Jolla, California 92093, United States

[§]A9.com, Palo Alto, California 94301, United States

[∥]Department of Chemistry, University of Florida, Gainesville, Florida 32611, United States

[⊥]GlaxoSmithKline PLC, 1250 South Collegeville Road, Collegeville, Pennsylvania 19426, United States

ABSTRACT: We report progress in graphics processing unit (GPU)-accelerated molecular dynamics and free energy methods in Amber18. Of particular interest is the development of alchemical free energy algorithms, including free energy perturbation and thermodynamic integration methods with support for nonlinear soft-core potential and parameter interpolation transformation pathways. These methods can be used in conjunction with enhanced sampling techniques such as replica exchange, constant-pH molecular dynamics, and new 12−6−4 potentials for metal ions. Additional performance enhancements have been made that enable appreciable speed-up on GPUs relative to the previous software release.

INTRODUCTION

Molecular simulation provides an extremely powerful tool for the interpretation of experimental data, the understanding of biomolecular systems, and the prediction of properties important for molecular design. As the scope and scale of applications increase, so must the computing capability of molecular simulation software. The past decade has seen rapid advances motivated by the performance enhancements offered by the latest molecular dynamics software packages written for specialized hardware. Perhaps the most affordable and impactful of these are platforms using graphics processing units (GPUs).[1−7]

The present application note reports on advances made in the latest release of the AMBER molecular simulation software suite (Amber18),[8] in particular enhancements of the primary GPU-accelerated simulation engine (PMEMD). These advancements significantly improve the program's execution of molecular simulations and offer new, integrated features for calculating alchemical free energies,[9,10] including thermodynamic integration (TI),[11−15] free energy perturbation (FEP),[15−19] and Bennett's acceptance ratio and its variants (BAR/MBAR),[20−25] as well as enhanced sampling, constant-pH simulation,[26,27] and the use of new 12−6−4 potentials.[28−30] Amber18 offers a broad solution for a wide range of free energy simulations, with expanded capability to compute forces in a hybrid environment of CPUs and GPUs, and establishes an add-on system for applying additional potential functions computed on the graphics processor without affecting optimizations of the main kernels. When run exclusively on GPU hardware, Amber18 shows consistent performance increases of 10% to 20% compared with Amber16 across Pascal (GTX-1080TI, Titan-XP, P100) and Volta architectures when running standard MD simulations, with more dependence on the system size than the architecture. Below we provide an overview of the software design, a description of new features, and performance benchmarks.

SOFTWARE DESIGN

Encapsulated Free Energy Modules. The development of molecular simulation software designed for optimal performance on specialized hardware requires customization and careful redesign of the underlying algorithmic implementation. In the case of the current GPU consumer market, single-precision floating-point (SPFP) performance outstrips that of double-precision performance by a significant amount. In order to address this issue in AMBER, new precision models[6,31] have been developed that leverage fixed-point integer arithmetic to replace slow double-precision arithmetic in certain key steps when higher precision is required, such as the accumulation of components of the force. Free energy simulations, because of the mixing/interpolation of hybrid

Received: July 12, 2018
Published: September 10, 2018

DOI: 10.1021/acs.jcim.8b00462

Hamiltonians and the averaging of TI and FEP quantities, present new challenges with respect to precision compared with conventional MD simulation. The GPU-enhanced features in Amber18 were designed to address these different precision requirements in order to ensure that statistical and thermodynamically derived properties are indistinguishable from those obtained using the CPU version of the code[32] while maintaining or improving the level of performance of previous versions of AMBER for conventional MD simulations. To fulfill these goals, we utilized two architectural concepts of object-oriented programming: encapsulation and inheritance.[33]The original AMBER GPU data structures are encapsulated into base C++ classes containing all coordinates, forces, energy terms, all simulation parameters, and settings. New free energy classes are derived from the base classes that contain the original GPU functionality and data structures for MD simulations. New derived free energy classes inherit all of the properties and methods of existing MD classes. Through encapsulation and inheritance, free energy capability can be implemented so that (1) there is little or no need to modify the original MD GPU codes except they are wrapped into base classes now, since new add-ons can be implemented in the derived free energy classes; (2) the new specific free energy algorithms and associated data structures are transparent to the base classes, such that modifying or optimizing the base classes will have minimal effects on the derived classes; and (3) derived free energy classes can utilize different algorithms, different precision models, and even different force fields.

Such an encapsulation and inheritance approach, on the other hand, could introduce additional computational overhead compared with direct modification of MD GPU kernels so that similar computational tasks are executed within the same kernels. Consequently the approach reported here will be ideal for new modules where only small portions of calculations need to be altered, such as TI, while direct modification of MD GPU kernels will be suitable for situations where algorithm changes are global, such as incorporation of polarizable force fields.

Extensions of New Modules. The present software design can be easily extended to accommodate implementation of new methods or algorithms. For example, the 12−6−4 potential modules have been implemented using this framework by treating the extra r^{-4} terms through add-on modules with minimal, if any, modification of MD CPU codes and GPU kernels.

For most developers, the CPU code remains the most accessible means for prototyping new methods. In selected cases where complex potentials are applied to only a small number of atoms, the CPU may afford performance advantages for development code that have to be fully optimized. To serve these needs, AMBER18 has a new "atom shuttling" system that extracts information on a predefined list of selected atoms and transfers it between the host CPU memory and GPU device. In previous versions, all coordinates, forces, charges, and other data can be downloaded and uploaded between the host and device at costs approaching 30% of the typical time step. When the majority of the system's atoms will not influence the CPU computations, this is wasteful. The shuttle system accepts a predefined list of atoms and collects them into a buffer for a lighter communication requirement between the host and device. The cost of organizing the atoms into the list is minor, but the extra kernel calls to manage the list and the latency of initiating the transfer are considerations. In the limit of transferring very small numbers of atoms, the cost of the shuttle can be less than 1% of the total simulation time, but methods that require transferring the majority of the atom data may be more efficient porting entire arrays with the methods in Amber16.

FEATURES

The current Amber18 has a host of features available that work together to perform MD and free energy simulations (Table 1). Brief descriptions of the most relevant features are provided below.

Table 1. Comparison of Free Energy (FEP/TI)-Compatible Features in Amber16 and Amber18 on CPUs and GPUs[a]

Free energy compatible features		Amber16		Amber18	
Category	Functionality	CPU	GPU	CPU	GPU
Ensemble	NVE	✓	✗	✓	✓
	NVT	✓	✗	✓	✓
	NPT	✓	✗	✓	✓
	semi-iso P	✗	✗	✗	✗
	CpHMD	✓	✗	✓	✓
Free Energy Analysis	TI	✓	✗	✓	✓
	MBAR	✓	✗	✓	✓
Enhanced Sampling	H-REMD	✓	✗	✓	✓
	AMD	✓	✗	✓	✓
	SGLD	✓	✗	✓	✗
	GAMD	✓	✗	✓	✓
Potentials	12-6-4	✓	✗	✓	✓

[a] List of features that have compatibility with free energy (FEP/TI) methods in Amber16 and Amber18 on CPUs and GPUs. Red color indicates a feature not compatible with free energy methods (although it may be compatible with conventional MD). Green color indicates new free-energy-compatible feature in Amber18.

Direct Implementation of Alchemical Free Energy Methods. Alchemical free energy simulations[9,10] provide accurate and robust estimates of relative free energies from molecular dynamics simulations[10,15,34−38] but are computationally intensive and are often limited by the availability of computing resources and/or required turnaround time. The limitations can render these methods impractical, particularly for industrial applications.[39] GPU-accelerated alchemical free energy methods change this landscape but have only recently emerged in a few simulation codes. The free energy methods implemented in the Amber18 GPU code build on the efficient AMBER GPU MD code base (pmemd.cuda) and include both TI and FEP classes.

In TI,[11−15] the free energy change from state 0 to state 1, $\Delta A_{0\rightarrow 1}$, is approximately calculated by numerical integration of the derivative of the system potential energy U with respect to the target parameter λ:

$$\Delta A_{0\rightarrow 1} = \int_0^1 \left\langle \frac{\mathrm{d}U(\lambda, \mathbf{q})}{\mathrm{d}\lambda} \right\rangle_{\lambda} \mathrm{d}\lambda \qquad (1)$$

In FEP,[15−19] the free energy change between state 0 and 1, $\Delta A_{0\rightarrow 1}$ is calculated by averaging the exponential of the potential energy differences sampled on the potential surface of state 0 (the Zwanzig equation[16]):

$$\Delta A_{0\to 1} = \beta^{-1}\ln\langle \mathrm{e}^{-\beta(U_1(\mathbf{q})-U_0(\mathbf{q}))}\rangle_0 = \beta^{-1}\ln\langle \mathrm{e}^{-\beta\Delta U(\mathbf{q})}\rangle_0 \qquad (2)$$

The quantities calculated from FEP can be output for postanalysis through Bennett's acceptance ratio and its variants (BAR/MBAR).[20−25]

Both TI and FEP methods can be used for linear alchemical transformations, nonlinear "parameter-interpolated" pathways[40] and so-called "soft core"[41−43] schemes for both van der Waals and electrostatic interactions. All of the above are available in the current Amber18 GPU release by utilizing the same input formats as the CPU version.

The GPU free energy implementation has been demonstrated to deliver speed increases of generally significantly more than 1 order of magnitude when a single GPU is compared with a comparably priced single (multicore) microprocessor (see Performance for detailed benchmarks). The GPU free energy implementation code performs TI with linear alchemical transformations roughly at the speed of 70% of running an MD simulation with the fast SPFP precision mode,[31] similar to the ratios seen in the CPU counterpart.[32] Overall, the free energy simulation speed-up relative to the CPU code is very similar to that for conventional MD simulation. As will be discussed in the next section, in certain instances the overhead (relative to conventional MD) for the linear alchemical free energy simulations can be greatly reduced by the use of nonlinear parameter-interpolated TI.[40]

Parameter-Interpolated Thermodynamic Integration. Amber18 is also able to exploit the properties of a parameter-interpolated thermodynamic integration (PI-TI) method,[40] which has recently been extended to support particle mesh Ewald (PME) electrostatics, to connect states by their molecular mechanical parameter values. This method has the practical advantage that no modification to the MD code is required to propagate the dynamics, and unlike with linear alchemical mixing, only one electrostatic evaluation is needed (e.g., a single call to PME). In the case of Amber18, this enables all of the performance benefits of GPU acceleration to be realized in addition to unlocking the full spectrum of features available within the MD software. The TI evaluation can be accomplished in a postprocessing step by reanalyzing the statistically independent trajectory frames in parallel for high throughput. Additional tools to streamline the computational pipeline for free energy postprocessing and analysis are forthcoming.

Replica-Exchange Molecular Dynamics. During the past two decades, the replica-exchange molecular dynamics (REMD) methods[44,45] have become popular in overcoming the multiple-minima problem by exchanging noninteracting replicas of the system under different conditions. The original replica-exchange methods were applied to systems at several temperatures[44] and have been extended to various conditions, such as Hamiltonian,[46] pH,[47] and redox potentials. Amber18 is capable of performing temperature, Hamiltonian, and pH replica-exchange simulations using the GPU. Hamiltonian replica exchange can be configured in a flexible way as long as the "force field" (or, equivalently, the prmtop file) is properly defined for each replica. Hence, the newly implemented free energy methods in Amber18 can be performed as Hamiltonian replica exchange so that different λ windows can exchange their conformations. Other types of Hamiltonian replica-exchange simulations, such as Hamiltonian tempering or umbrella sampling, can be easily set up as well.

Multiple-dimension replica-exchange simulations,[48−53] in which two or more conditions are simulated at the same time, are supported as well. By the use of the multidimensional replica-exchange capability, many practical combinations are possible, such as TI simulation combined with temperature or pH replica exchange.

The configuration of GPUs in Amber18 replica-exchange simulations is very flexible in order to fit various types of computational resources. Ideally for load balancing, the number of replicas should be an integer multiple (typically 1−6) of the number of available GPUs. One GPU can run one or multiple replicas if sufficient GPU memory is available, although one can expect some slowdown in cases where multiple tasks are running concurrently on a single GPU. Our experience has shown that an 11 GB GTX 1080TI GPU can handle six instances of typical kinase systems (around 30 000 to 50 000 atoms) without losing efficiency. One scenario is that to run free energy simulations with replica exchange on one multiple GPU node, e.g., executing 12 λ windows on a four- or six-GPU node with each GPU handling three or two λ windows, respectively. Such scenarios take advantage of extremely fast intranode communication and enable efficient performance optimization on modern large-scale GPU clusters/supercomputers such Summit at Oak Ridge National Laboratory. In principle, a single replica can also be run in parallel on multiple GPUs, but this is strongly discouraged because Amber18 is not optimized for it.

Constant-pH Molecular Dynamics. Conventional all-atom molecular simulations consider ensembles constrained to have predetermined fixed protonation states that are not necessarily consistent with any pH value. Constant-pH molecular dynamics (CpHMD) is a technique that enables sampling of different accessible protonation states (including different relevant tautomers) consistent with a bulk pH value.[26,27] These methods have been applied to a wide array of biological problems, including prediction of pK_a shifts in proteins and nucleic acids and pH-dependent conformational changes, assembly, and protein−ligand, protein−protein, and protein−nucleic acid binding events.[54] These methods provide detailed information about the conditional probability of observing correlated protonation events that have biological implications. Very recently, a discrete-protonation-state CpHMD method has been implemented on GPUs, integrated with REMD methods (including along a pH dimension), and tested in AMBER.[55] The method was applied for the first time to the interpretation of activity−pH profiles in a mechanistic computational enzymology study of the archetype enzyme RNase A.[56] The CpHMD method in Amber18 is compatible with enhanced sampling methods such as REMD and is compatible with the new GPU-accelerated free energy framework.

The workflow of explicit-solvent CpHMD simulation has been described in detail elsewhere.[55] Briefly, the method follows the general approach of Baptista and co-workers[26,27] that involves sampling of discrete protonation states using a Monte Carlo sampling procedure. Simulations are performed in explicit solvent under periodic boundary conditions using PME to generate ensembles. The CpHMD method utilizes an extended force field that contains parameters (typically charge vectors) associated with changes in protonation state and reference chemical potentials for each titratable site calibrated for a selected generalized Born (GB) model to obtain correct pK_a values in solution. In the Monte Carlo decision to accept or reject a trial protonation state, explicit solvent (including any nonstructural ions) is stripped and replaced using the selected GB model under nonperiodic boundary conditions. Additional considerations are made for multisite titration involving titratable residues that are considered to be "neighbors".[55] If any protonation state change attempts are accepted, the explicit solvent is replaced, the solute is frozen, and MD is used to relax the solvent degrees of freedom for a short period of time. After relaxation is complete, the velocities of the solute atoms are restored to their prior values and standard dynamics resumes. Full details can be found in ref 55.

12−6−4 Potentials for Metal Ions. The GPU version of Amber18 (pmemd.cuda) is capable of utilizing 12−6−4 potentials, which were developed by Li et al.[28] for metal ions in aqueous solution and recently extended for Mg^{2+}, Mn^{2+}, Zn^{2+}, and Cd^{2+} ions so as to have balanced interactions with nucleic acids.[30] The 12−6−4 potentials are derived from regular Lennard-Jones (LJ) 12−6 potentials by adding r^{-4} terms, as proposed by Roux and Karplus.[57,58]

The 12−6 potential[59] for nonbonded interactions is

$$U_{ij}(r_{ij}) = \epsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{R_{ij}}{r_{ij}}\right)^{6}\right] \tag{3}$$

where the parameters R_{ij} and ϵ_{ij} are the combined radius and well depth for the pairwise interaction, respectively, and r_{ij} is the distance between the particles. Equation 3 can be expressed equivalently as

$$U_{ij}(r_{ij}) = \frac{A_{ij}}{r_{ij}^{\ 12}} - \frac{B_{ij}}{r_{ij}^{\ 6}} \tag{4}$$

where $A_{ij} = \epsilon_{ij}R_{ij}^{\ 12}$ and $B_{ij} = 2\epsilon_{ij}R_{ij}^{\ 6}$. The expanded 12−6−4 potential[60] is then

$$\begin{aligned} U_{ij}(r_{ij}) &= \epsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{R_{ij}}{r_{ij}}\right)^{6} - 2\kappa R_{ij}^{\ 2}\left(\frac{R_{ij}}{r_{ij}}\right)^{4}\right] \\ &= \frac{A_{ij}}{r_{ij}^{\ 12}} - \frac{B_{ij}}{r_{ij}^{\ 6}} - \frac{C_{ij}}{r_{ij}^{\ 4}} \end{aligned} \tag{5}$$

where $C_{ij} = B_{ij}\kappa$ and κ is a scaling parameter with units of Å^{-2}. The additional attractive term, $-C_{ij}/r_{ij}^{\ 4}$, implicitly accounts for polarization effects by mimicking the charge-induced dipole interaction. The 12−6−4 potentials have showed a marked improvement over the LJ 12−6 nonbonded model.[28,30]

Future Plan. The Amber18 development roadmap will extend sampling capabilites for free energy simulations to facilitate advancement of drug discovery,[61] including implementation of the Gibbs sampling scheme[62] to improve the exchange rates in replica-exchange simulations, self-adjusted mixture sampling (SAMS)[63] to optimize the simulation lengths for different λ windows, and replica exchange with solute scaling (REST2),[64] a scheme to more stably and efficiently perform "effective" solute-tempering[65] replica-exchange simulations.

■ PERFORMANCE

Amber18 runs efficiently on GPU platforms for both MD and free energy simulations. Performance benchmarks for equilibrium MD are shown in Figure 1 and listed for selected GPUs in Table 2, including comparisons with Amber16. The figure

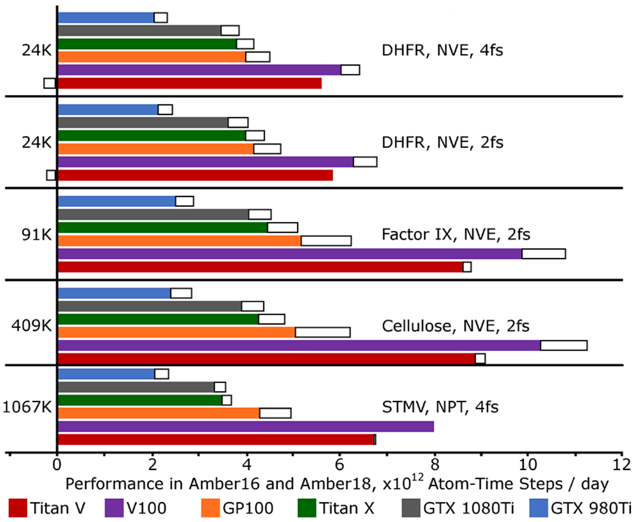

Figure 1. Performance of Amber18 relative to Amber16 seen on multiple GPU architectures. Performance is given in a particle-normalized metric that emphasizes the number of interactions that each card is able to compute in a given time. Performances in Amber16 are shown as solid color bars and improvements with Amber18 as black-outlined extensions. In a few cases, the performance in Amber18 is lower than in Amber16, as indicated by placement of the extensions to the left of the y axis. (Beta tests of an upcoming patch make Amber18 even faster and consistently superior to Amber16.) The systems, ensembles, and time steps are displayed at the right, while the system sizes (in thousands of atoms) are given at the left. All of the systems were run with an 8 Å cutoff for real-space interactions and other default Amber parameters.

works in a particle-normalized metric, trillions of atom-time steps per day, which puts most systems on equal footing and shows performance improvements of up to 24% without implying improper comparisons to other codes (the cutoffs used in these benchmarks are smaller than those in some other benchmarks, and other settings may not be comparable). Longer time steps, if safe, tend to improve the overall throughput with a marginal increase in the cost of computing each step (requiring more frequent pair list updates). Small systems tend to perform less efficiently (small FFTs and pair list building kernels do not fully occupy the GPU). Virial computations are also costly, as seen for the Satellite Tobacco Mosaic Virus (STMV) system, the only one of this abbreviated list of benchmarks to include pressure regulation with a Berendsen barostat.

On a GTX-1080Ti, still the most cost-effective GPU at the time of publication, the 23 558 atom dihydrofolate reductase (DHFR) benchmark (4 fs time step, constant energy) runs at 657 ns/day in Amber18 and 588 ns/day in Amber16. The same codes run the 90 906 atom blood-clotting factor IX system at 100 and 89 ns/day, respectively, with a 2 fs time step. The performance in thermodynamic integration free energy simulations for mutating ligands of the clotting factor Xa system is shown in Figure 2. TI with linear alchemical mixing generally exacts a toll of one-third of the speed that could be achieved in a conventional MD simulation. Additional pairwise computations between particles are present, but the secondary reciprocal-space calculation is about 85% of the additional cost (this cost is eliminated in the PI-TI method[40]). The main

Table 2. Comparison of MD Simulation Rates in Amber16 and Amber18 on CPUs and GPUs[a]

		simulation rate, ns/day					
		GTX-980 Ti		GTX-1080 Ti		Titan-X	
system	atom count	Amber16	Amber18	Amber16	Amber18	Amber16	Amber18
DHFR, *NVE*, 4 fs	24k	347	382	588	657	643	710
DHFR, *NVE*, 2 fs	24k	181	209	306	345	338	374
factor IX, *NVE*, 2 fs	91k	52	64	85	100	93	113
cellulose, *NVE*, 2 fs	409k	12	14	19	21	21	23
STMV, *NVE*, 4 fs	1067k	8	9	12	13	13	14
		Simulation Rate, ns/day					
		GP100		V100 (Volta)		Titan-V	
system	atom count	Amber16	Amber18	Amber16	Amber18	Amber16	Amber18
DHFR, *NVE*, 4 fs	24k	677	768	1020	1091	954	904
DHFR, *NVE*, 2 fs	24k	353	404	532	577	497	477
factor IX, *NVE*, 2 fs	91k	114	137	217	238	189	194
cellulose, *NVE*, 2 fs	409k	25	29	50	49	43	41
STMV, *NVE*, 4 fs	1067k	16	19	30	30	25	25

[a] Timings for selected systems in Amber18 versus Amber16 are shown. The ensemble and time step are given in the first column. Other Amber default parameters included an 8 Å cutoff and a ≤1 Å PME (3D-FFT) grid spacing.

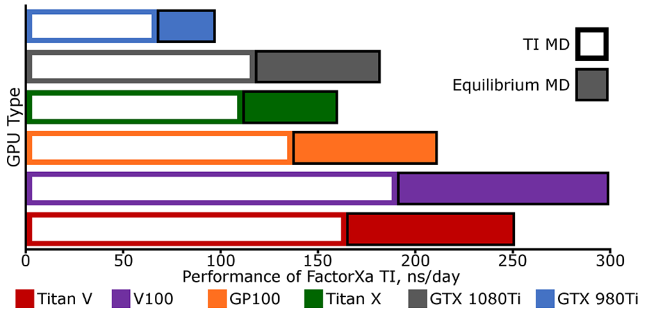

Figure 2. Performance of Amber18 thermodynamic integration with linear alchemical mixing on multiple GPU architectures relative to conventional MD. The color scheme for each GPU type is consistent with Figure 1, but the performance of TI is given by an open bar while the performance of the equivalent "plain" MD system is given by a black-bordered solid extension. The test system is the factor Xa protein with the ligand mutation from L51a (62 atoms) to L51b (62 atoms). The system has a total of 41 563 atoms, and the whole ligand is defined as the TI region.

performance improvements derive from (1) innovative spline tabulation lookup and particle mapping kernels for faster PME direct and reciprocal-space calculations and (2) more efficient memory access for bonded and nonbonded terms.

Faster PME Direct and Reciprocal-Space Calculations. Most CPU codes use a quadratic or cubic spline for the derivative of the complementary error function used in the PME direct-space energy term. Rather than costly conversion of floating-point values into integer indexes for table lookups, we take the IEEE-754 representation of the 32-bit floating point number for the squared distance and use its high 14 bits, isolated by interpreting it as an unsigned integer and shifting right 18 bits, as an integer index into a logarithmically coarsening lookup table. This approach uses a minimum of the precious streaming multiprocessor (SMP) cache, collects a huge number of arithmetic operations into a single cubic spline evaluation, and typically leads to a 6−8% speedup. The workflow of the nonbonded kernel was further improved by eliminating _shared_ memory storage and dealing with all particle comparisons within the same warp via _shfl instructions. This permitted us to engage not just 768 but 1280 threads on each SMP.

PME Reciprocal Space. We have made improvements to the kernel that maps particles onto the 3DFFT mesh by parallel computation of *B*-spline coefficients for all three dimensions (utilizing 90 out of 96 threads in the block rather than less than one-third of them) and retuning the stencil for writing data onto the mesh to make better-coalesced atomic transactions. This improves the throughput of the mapping kernel by more than 40% and typically leads to a few percent speedup overall.

More Efficient Memory Access for Bonded and Nonbonded Terms. Rather than reach into global memory for the coordinates of each individual atom needed by any bonded term, we draw groups of topologically connected atoms at the start of the simulation and assign bond and angle terms to operate on the atoms of these groups. At each step of the simulation, the coordinates of each group are cached on the SMP, and forces due to their bonded interactions are accumulated in _shared_ memory. Last, the results are dumped back to global via atomic transactions, reducing the global reads and writes due to bonded interactions more than 10-fold. The approach generalizes one described 10 years ago for the Folding@Home client,[66] where bond and angle computations were computed by the threads that had already downloaded the atoms for a dihedral computation. Our approach makes much larger groups of atoms (up to 128) and does not compute redundant interactions. However, the block-wide synchronization after reading coordinates and prior to writing results may leave threads idle. The modular programming that creates our networks of interactions facilitates combining or partitioning the GPU kernels to optimize register usage and thread occupancy.

We have also gained a considerable amount of improvement by trimming the precision model where low significant bits are wasted. Rather than convert every nonbonded force to 64-bit integers immediately, we accumulate forces from 512 interactions (evaluated sequentially in sets of 16 by each of 32 threads in a warp) before converting the sums to integer and ultimately committing the result back to global memory. Because the tile scheme in our nonbonded kernel remains

DOI: 10.1021/acs.jcim.8b00462

warp-synchronous, the sequence of floating-point operations that evaluates the force on each atom is identical regardless of the order the tile was called. In other words, each tile evaluation is immune from race conditions. Conversions to integer arithmetic always occur, as in Amber16, before the results of separate warps are combined, as the coordination of different warps is not guaranteed. These optimizations therefore maintain the numerical determinism of the code: a given GPU will produce identical answers for a given system and input parameters.

Minimal Computational Load on CPU and GPU/CPU Intercommunication. As it does for MD, Amber18 performs free energy computations almost entirely on the GPU and requires very little communication between the CPU and the GPU. This results in a tremendous practical advantage over other implementations in that independent or loosely coupled simulations (e.g., different λ windows of a TI or FEP, possibly with REMD) can be run efficiently in parallel on cost-effective nodes that contain multiple GPUs with a single (possibly low-end) CPU managing them all without loss of performance. This is a critical design feature that distinguishes Amber18 free energy simulations from other packages that may require multiple high-end CPU cores to support each GPU for standard dynamics and free energy calculations. The result is an implementation of TI/FEP that is not only one of the fastest available but also the most cost-effective when hardware costs are factored in.

■ CONCLUSION

In this application note, we have reported new features and performance benchmarks for the Amber18 software official release. The code is able to perform GPU-accelerated alchemical free energy perturbation and thermodynamic integration highly efficiently on a wide range of GPU hardware. The free energy perturbation simulations output metadata that can be analyzed using conventional or multistate Bennett's acceptance ratio methods. Additionally, thermodynamic integration capability is enabled for linear alchemical transformations and nonlinear transformations including soft-core potentials and parameter-interpolated TI methods recently extended for efficient use with particle mesh Ewald electrostatics. These free energy methods can be used in conjunction with a wide range of enhanced sampling methods, constant-pH molecular dynamics, and new 12−6−4 potentials for metal ions. The Amber18 software package provides a rich set of high-performance GPU-accelerated features that enable a wide range of molecular simulation applications from computational molecular biophysics to drug discovery.

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: ross@rosswalker.co.uk.
*E-mail: Darrin.York@rutgers.edu.

ORCID

Tai-Sung Lee: 0000-0003-2110-2279
Charles Lin: 0000-0002-5461-6690
Adrian Roitberg: 0000-0003-3963-8784
David A. Case: 0000-0003-2314-2346
Darrin M. York: 0000-0002-9193-7055

Notes

The authors declare no competing financial interest.

Amber18 is available for download from the AMBER home page: http://ambermd.org/.

■ ACKNOWLEDGMENTS

The authors are grateful for financial support provided by the National Institutes of Health (GM107485 and GM62248 to D.M.Y.). Computational resources were provided by the Office of Advanced Research Computing (OARC) at Rutgers, The State University of New Jersey, the National Institutes of Health under Grant S10OD012346, the Blue Waters Sustained-Petascale Computing Project (NSF OCI 07-25070 and PRAC OCI-1515572), and the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation (ACI-1548562 and OCI-1053575). We gratefully acknowledge the support of the nVidia Corporation with the donation of several Pascal and Volta GPUs and the GPU time on the GPU cluster where the reported benchmark results were performed.

■ REFERENCES

(1) Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Hardy, D. J.; Trabuco, L. G.; Schulten, K. Accelerating molecular modeling applications with graphics processors. *J. Comput. Chem.* **2007**, *28*, 2618−2640.

(2) Anderson, J. A.; Lorenz, C. D.; Travesset, A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.* **2008**, *227*, 5342−5359.

(3) Harvey, M. J.; Giupponi, G.; Fabritiis, G. D. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *J. Chem. Theory Comput.* **2009**, *5*, 1632−1639.

(4) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. GPU-accelerated molecular modeling coming of age. *J. Mol. Graphics Modell.* **2010**, *29*, 116−125.

(5) Xu, D.; Williamson, M. J.; Walker, R. C. Chapter 1 - Advancements in Molecular Dynamics Simulations of Biomolecules on Graphical Processing Units. *Annu. Rep. Comput. Chem.* **2010**, *6*, 2−19.

(6) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9*, 3878−3888.

(7) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.* **2017**, *13*, e1005659.

(8) Case, D. A.; et al. *AMBER 18*; University of California: San Francisco, 2018.

(9) Straatsma, T. P.; McCammon, J. A. Computational alchemy. *Annu. Rev. Phys. Chem.* **1992**, *43*, 407−435.

(10) *Free Energy Calculations: Theory and Applications in Chemistry and Biology*; Chipot, C., Pohorille, A., Eds.; Springer Series in Chemical Physics, Vol. *86*; Springer: New York, 2007.

(11) Kirkwood, J. G. Statistical mechanics of fluid mixtures. *J. Chem. Phys.* **1935**, *3*, 300−313.

(12) Straatsma, T. P.; Berendsen, H. J. C.; Postma, J. P. M. Free energy of hydrophobic hydration: A molecular dynamics study of noble gases in water. *J. Chem. Phys.* **1986**, *85*, 6720−6727.

(13) Straatsma, T. P.; Berendsen, H. J. Free energy of ionic hydration: Analysis of a thermo-dynamic integration technique to evaluate free energy differences by molecular dynamics simulations. *J. Chem. Phys.* **1988**, *89*, 5876−5886.

(14) Straatsma, T. P.; McCammon, J. A. Multiconfiguration thermodynamic integration. *J. Chem. Phys.* **1991**, *95*, 1175−1188.

(15) Shirts, M. R.; Pande, V. S. Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett

acceptance ratio, and thermodynamic integration. *J. Chem. Phys.* **2005**, *122*, 144107.

(16) Zwanzig, R. W. High-temperature equation of state by a perturbation method. I. Non-polar gases. *J. Chem. Phys.* **1954**, *22*, 1420−1426.

(17) Torrie, G. M.; Valleau, J. P. Monte Carlo free energy estimates using non-Boltzmann sampling: Application to the sub-critical Lennard-Jones fluid. *Chem. Phys. Lett.* **1974**, *28*, 578−581.

(18) Jarzynski, C. Equilibrium free-energy differences from nonequilibrium measurements:a master-equation approach. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **1997**, *56*, 5018−5035.

(19) Lu, N.; Kofke, D. A. Accuracy of free-energy perturbation calculations in molecular simulation. II. Heuristics. *J. Chem. Phys.* **2001**, *115*, 6866−6875.

(20) Bennett, C. H. Efficient estimation of free energy differences from Monte Carlo data. *J. Comput. Phys.* **1976**, *22*, 245−268.

(21) Shirts, M. R.; Bair, E.; Hooker, G.; Pande, V. S. Equilibrium free energies from nonequilibrium measurements using maximum-likelihood methods. *Phys. Rev. Lett.* **2003**, *91*, 140601.

(22) Shirts, M. R.; Chodera, J. D. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* **2008**, *129*, 124105.

(23) Konig, G.; Boresch, S. Non-Boltzmann sampling and Bennett's acceptance ratio method: how to profit from bending the rules. *J. Comput. Chem.* **2011**, *32*, 1082−1090.

(24) Mikulskis, P.; Genheden, S.; Ryde, U. A large-scale test of free-energy simulation estimates of protein-ligand binding affinities. *J. Chem. Inf. Model.* **2014**, *54*, 2794−2806.

(25) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. Guidelines for the analysis of free energy calculations. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 397−411.

(26) Baptista, A. M.; Teixeira, V. H.; Soares, C. M. Constant-pH molecular dynamics using stochastic titration. *J. Chem. Phys.* **2002**, *117*, 4184−4200.

(27) Chen, J.; Brooks, C. L., III; Khandogin, J. Recent advances in implicit solvent-based methods for biomolecular simulations. *Curr. Opin. Struct. Biol.* **2008**, *18*, 140−148.

(28) Li, P.; Roberts, B. P.; Chakravorty, D. K.; Merz, K. M., Jr. Rational design of Particle Mesh Ewald compatible Lennard-Jones parameters for + 2 metal cations in explicit solvent. *J. Chem. Theory Comput.* **2013**, *9*, 2733−2748.

(29) Panteva, M. T.; Giambaşu, G. M.; York, D. M. Comparison of structural, thermodynamic, kinetic and mass transport properties of Mg^{2+} ion models commonly used in biomolecular simulations. *J. Comput. Chem.* **2015**, *36*, 970−982.

(30) Panteva, M. T.; Giambasu, G. M.; York, D. M. Force field for Mg^{2+}, Mn^{2+}, Zn^{2+}, and Cd^{2+} ions that have balanced interactions with nucleic acids. *J. Phys. Chem. B* **2015**, *119*, 15460−15470.

(31) Le Grand, S.; Götz, A. W.; Walker, R. C. SPFP: Speed without compromise—A mixed precision model for GPU accelerated molecular dynamics simulations. *Comput. Phys. Commun.* **2013**, *184*, 374−380.

(32) Lee, T.-S.; Hu, Y.; Sherborne, B.; Guo, Z.; York, D. M. Toward Fast and Accurate Binding Affinity Prediction with pmemdGTI: An Efficient Implementation of GPU-Accelerated Thermodynamic Integration. *J. Chem. Theory Comput.* **2017**, *13*, 3077−3084.

(33) Scott, M. L. *Programming Language Pragmatics*; Morgan Kaufmann Publishers: San Francisco, 2000.

(34) Chodera, J.; Mobley, D.; Shirts, M.; Dixon, R.; Branson, K.; Pande, V. Alchemical free energy methods for drug discovery: progress and challenges. *Curr. Opin. Struct. Biol.* **2011**, *21*, 150−160.

(35) Gallicchio, E.; Levy, R. M. Advances in all atom sampling methods for modeling protein-ligand binding affinities. *Curr. Opin. Struct. Biol.* **2011**, *21*, 161−166.

(36) Bruckner, S.; Boresch, S. Efficiency of alchemical free energy simulations. I. A practical comparison of the exponential formula, thermodynamic integration, and Bennett's acceptance ratio method. *J. Comput. Chem.* **2011**, *32*, 1303−1319.

(37) Hansen, N.; van Gunsteren, W. F. Practical Aspects of Free-Energy Calculations: A Review. *J. Chem. Theory Comput.* **2014**, *10*, 2632−2647.

(38) Homeyer, N.; Stoll, F.; Hillisch, A.; Gohlke, H. Binding Free Energy Calculations for Lead Optimization: Assessment of Their Accuracy in an Industrial Drug Design Context. *J. Chem. Theory Comput.* **2014**, *10*, 3331−3344.

(39) Chipot, C.; Rozanska, X.; Dixit, S. B. Can free energy calculations be fast and accurate at the same time? Binding of low-affinity, non-peptide inhibitors to the SH2 domain of the src protein. *J. Comput.-Aided Mol. Des.* **2005**, *19*, 765−770.

(40) Giese, T. J.; York, D. M. A GPU-Accelerated Parameter Interpolation Thermodynamic Integration Free Energy Method. *J. Chem. Theory Comput.* **2018**, *14*, 1564−1582.

(41) Steinbrecher, T.; Mobley, D. L.; Case, D. A. Nonlinear scaling schemes for Lennard-Jones interactions in free energy calculations. *J. Chem. Phys.* **2007**, *127*, 214108.

(42) Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. *Chem. Phys. Lett.* **1994**, *222*, 529−539.

(43) Steinbrecher, T.; Joung, I.; Case, D. A. Soft-Core Potentials in Thermodynamic Integration: Comparing One- and Two-Step Transformations. *J. Comput. Chem.* **2011**, *32*, 3253−3263.

(44) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141−151.

(45) Earl, D. J.; Deem, M. W. Parallel tempering: theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3910−3916.

(46) Arrar, M.; de Oliveira, C. A. F.; Fajer, M.; Sinko, W.; McCammon, J. A. w-REXAMD: A Hamiltonian replica exchange approach to improve free energy calculations for systems with kinetically trapped conformations. *J. Chem. Theory Comput.* **2013**, *9*, 18−23.

(47) Meng, Y.; Roitberg, A. E. Constant pH Replica Exchange Molecular Dynamics in Biomolecules Using a Discrete Protonation Model. *J. Chem. Theory Comput.* **2010**, *6*, 1401−1412.

(48) Sugita, Y.; Okamoto, Y. Replica-exchange multicanonical algorithm and multicanonical replica-exchange method for simulating systems with rough energy landscape. *Chem. Phys. Lett.* **2000**, *329*, 261−270.

(49) Bergonzo, C.; Henriksen, N. M.; Roe, D. R.; Swails, J. M.; Roitberg, A. E.; Cheatham, T. E., III Multidimensional replica exchange molecular dynamics yields a converged ensemble of an RNA tetranucleotide. *J. Chem. Theory Comput.* **2014**, *10*, 492−499.

(50) Mitsutake, A.; Okamoto, Y. Multidimensional generalized-ensemble algorithms for complex systems. *J. Chem. Phys.* **2009**, *130*, 214105.

(51) Gallicchio, E.; Levy, R. M.; Parashar, M. Asynchronous replica exchange for molecular simulations. *J. Comput. Chem.* **2008**, *29*, 788−794.

(52) Radak, B. K.; Romanus, M.; Gallicchio, E.; Lee, T.-S.; Weidner, O.; Deng, N.-J.; He, P.; Dai, W.; York, D. M.; Levy, R. M.; Jha, S. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13)*; ACM: New York, 2013; Vol. *26*; p 26.

(53) Radak, B. K.; Romanus, M.; Lee, T.-S.; Chen, H.; Huang, M.; Treikalis, A.; Balasubramanian, V.; Jha, S.; York, D. M. Characterization of the Three-Dimensional Free Energy Manifold for the Uracil Ribonucleoside from Asynchronous Replica Exchange Simulations. *J. Chem. Theory Comput.* **2015**, *11*, 373−377.

(54) Chen, W.; Morrow, B. H.; Shi, C.; Shen, J. K. Recent development and application of constant pH molecular dynamics. *Mol. Simul.* **2014**, *40*, 830−838.

(55) Swails, J. M.; York, D. M.; Roitberg, A. E. Constant pH replica exchange molecular dynamics in explicit solvent using discrete protonation states: Implementation, testing, and validation. *J. Chem. Theory Comput.* **2014**, *10*, 1341−1352.

(56) Dissanayake, T.; Swails, J. M.; Harris, M. E.; Roitberg, A. E.; York, D. M. Interpretation of pH-Activity Profiles for Acid-Base Catalysis from Molecular Simulations. *Biochemistry* **2015**, *54*, 1307−1313.

(57) Roux, B.; Karplus, M. Ion transport in a model gramicidin channel. Structure and thermodynamics. *Biophys. J.* **1991**, *59*, 961−981.

(58) Roux, B.; Karplus, M. Potential energy function for cation−peptide interactions: An ab initio study. *J. Comput. Chem.* **1995**, *16*, 690−704.

(59) Jones, J. E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas. *Proc. R. Soc. London, Ser. A* **1924**, *106*, 463−477.

(60) Li, P.; Merz, K. M., Jr. Taking into account the ion-induced dipole interaction in the nonbonded model of ions. *J. Chem. Theory Comput.* **2014**, *10*, 289−297.

(61) Abel, R.; Wang, L.; Harder, E. D.; Berne, B. J.; Friesner, R. A. Advancing Drug Discovery through Enhanced Free Energy Calculations. *Acc. Chem. Res.* **2017**, *50*, 1625−1632.

(62) Chodera, J. D.; Shirts, M. R. Replica exchange and expanded ensemble simulations as Gibbs sampling: Simple improvements for enhanced mixing. *J. Chem. Phys.* **2011**, *135*, 194110.

(63) Tan, Z. Optimally Adjusted Mixture Sampling and Locally Weighted Histogram Analysis. *J. Comput. Graph. Stat.* **2017**, *26*, 54−65.

(64) Wang, L.; Friesner, R. A.; Berne, B. Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2). *J. Phys. Chem. B* **2011**, *115*, 9431−9438.

(65) Liu, P.; Kim, B.; Friesner, R. A.; Berne, B. J. Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 13749−13754.

(66) Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. Accelerating molecular dynamic simulation on graphics processing units. *J. Comput. Chem.* **2009**, *30*, 864−872.