The UA↔CG Workflow: High Performance Molecular Dynamics of Coarse-Grained Polymers

David Ozog[*], Allen D. Malony[*,‡],Marina Guenza[†,‡]
[*]*Department of Computer and Information Sciences*
[†]*Department of Chemistry and Biochemistry*
University of Oregon
Eugene, Oregon, USA
{*ozog, mguenza, malony*}*@uoregon.edu*

***Abstract*—Our analytically based technique for coarse-graining (CG) polymer simulations dramatically improves spatial and temporal scaling while preserving thermodynamic quantities and bulk properties. The purpose of CG codes is to run more efficient molecular dynamics simulations, yet the research field generally lacks thorough analysis of how such codes scale with respect to full-atom representations. This paper conducts an in-depth performance study of highly realistic polymer melts on modern supercomputing systems. We also present a workflow that integrates our analytical solution for calculating CG forces with new high-performance techniques for mapping back and forth between the atomistic and CG descriptions in LAMMPS. The workflow benefits from the performance of CG, while maintaining full-atom accuracy. Our results show speedups up to 12x faster than atomistic simulations.**

***Keywords*-atomistic simulation, coarse-graining, scientific workflows, polymeric liquids, LAMMPS**

I. Introduction

On the largest modern supercomputers, molecular dynamics (MD) simulations of polymer systems contain billions of atoms and span roughly a few nanoseconds of simulation time per week of execution time. Unfortunately, most macromolecular processes of interest contain many orders of magnitude more particles and often bridge microsecond or even millisecond timescales or longer. These include phenomena like phase separation in polymer blends and composite materials [1], polymer crystallization, and glass formation and aging [2] to mention just a few. Despite our pervasive access to massively parallel computers, full *united-atom (UA)* simulations do not come close to representing real-world polymer systems (see Figure 1), because they are too computationally expensive and slow. This makes direct comparison between experiments and simulations impossible as the two systems are dynamically different. For these reasons, scalability of MD simulations is paramount.

Simply put, we require new approximation methods that capture the relevant physics and chemistry while requiring fewer computational resources. The most promising approach is the *coarse-graining (CG)* method, in which groups of atoms are represented as one collective unit. CG has proven to be valuable for eliminating unnecessary degrees of freedom and tackling the scaling complexity of larger problems [3]. The key issue is how to simultaneously maintain solution accuracy and high performance. With the alternation of CG and atomistic simulations enabled by the workflow presented in this paper, it is possible to quickly equilibrate the system during the CG simulation, then reintroduce local details into a UA simulation, taking advantage of the performance of the CG simulation and the realism of the UA representation.

The *Integral Equation Coarse-Grained (IE-CG)* model by Guenza and coworkers [5], [6], [7], [8], [9] adopts an analytically-derived potential and dramatically improves spatial and temporal scaling of polymer simulations, while accurately preserving thermodynamic quantities and bulk properties [10], [11], [12]. Several numerical techniques and force fields exist for performing coarse-grained simulations [13], [14], [15]. However, these methods generally preserve either structure or fully preserve thermodynamics, but not both. As a result, only a small level of coarse-graining is typically adopted to limit the errors in the simulated structure and thermodynamics. In contrast, our work adopts the analytical approach offered by IE-CG theory, because it recovers crucial structural and thermodynamic quantities such as the equation of state, excess free energy, and pressure, while enabling a much higher level of coarse-graining and the corresponding gains in computational performance.

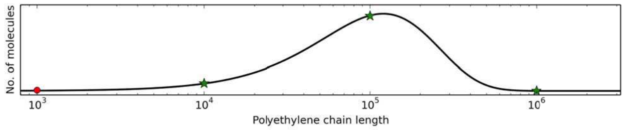

Figure 1: A representation of the average polyethylene chain length determined by chromatography experiments [4]. Most studies are limited to very short chain lengths (≤ 1000) due to the prohibitive cost of UA simulations, but this paper freely explores the realistic systems with 10^4 to 10^6 monomers per chain.

[‡] Co-corresponding authors

Although CG polymer physics is a mature field, little has been done to analyze the performance benefits of CG versus UA representations. While it is clear that CG will exhibit computational gains, does it strong scale to as many processors as the corresponding UA simulation? Likely not, because CG tracks far fewer overall particles, sometimes by orders of magnitude. Accordingly, the scalability of CG simulations likely depends on the granularity factor, e.g, the number of UA coordinates a CG unit represents.

One reason for the lack of performance analysis in CG research is likely due to the inherent complexity and variability in executing useful CG simulations. For instance, CG representations are generally based on a corresponding (usually unequilibrated) UA geometry. A helper program, which is usually independent of the MD simulation framework, then maps the UA representation into the CG representation. Furthermore, after the CG simulation equilibrates, we usually desire a "backmapped" geometric description of the equilibrated system in a *UA* representation to restore properties at the local molecular scale. The amalgamation of these processing steps encompass a scientific workflow for conduction CG simulations, shown pictorially in Figure 2. In order to benefit most from CG computational gains, the coupled processing stages of this workflow must be high performance, low overhead, and asynchronous whenever possible.

In this paper, we present such a scientific workflow that integrates the analytical IE-CG approach for calculating CG forces with new high-performance techniques for mapping back and forth between the UA and CG descriptions in LAMMPS. This workflow benefits from the performance of CG, while maintaining the accuracy of the full-atom representation. Our workflow optimizations legitimize our comparisons between UA and CG execution times. Scaling results show speedups of up to 12x at 3,072 cores on the Hopper system at NERSC. Furthermore, our workflow opens possibilities for the validation of polymeric systems that have never before been simulated at realistic chain lengths.

II. Background

This sections provides background information and context for understanding the motivation, design, and implementation of our scientific workflow that manages CG simulations of polymer melts. Section II-A describes our analytically based CG methods. Sections II-B and II-C examine the UA↔CG workflow and two crucial optimizations for assuring efficient execution.

A. Integral Equation Coarse-Graining

This section briefly reviews the Integral Equation Coarse-grained approach [5], [6], [7], [8], [9]. We simulate a homopolymer fluid (in which all monomers are of the same chemical type) with monomer number density, ρ_m, consisting of n polymer chains at temperature T. Each polymer contains N monomer sites. At the atomistic resolution, MD simulations were performed with LAMMPS using the UA model, where each site is either a CH, CH_2 or CH_3 group. In the coarse-grained representation, each polymer is described as a chain of soft particles, or spheres, and each sphere represents a number of N_b monomers. The total number of spheres is given by $n_b = N/N_b$. Lower numbers of spheres in the CG representation present lower computational resource requirements in the CG simulation.

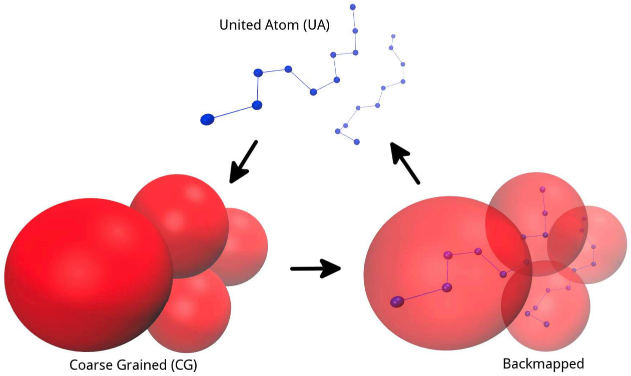

Figure 2: The high-level progression of a UA↔CG workflow. The CG representation is calculated from UA coordinates, and the UA representation is recovered by solving a backmapping problem (described in Sections II-C and III-B). CG spheres appear hard, but are soft with long range effects.

The IECG is an integral equation formalism that builds on the Ornstein-Zernike [16] equation and the PRISM atomistic theory [17]. The IECG model gives a complete description of the coarse-grained system as consisting of the effective intermolecular potential between coarse-grained units on different chains, effective bond potentials, and angle potentials designed to preserve Gaussian statistical distributions [18] and by postulating that the effective intermolecular potential must act between monomers farther apart on the same chain [10], [11], [12]. The intermolecular pair potential acting between CG units is fully represented as a function of the physical and molecular parameters that define the system, which are N_b, ρ_m, T and the liquid compressibility through the direct correlation function c_0.

In the specific regime of $N_b \geq 30$ it is possible to derive an analytical form of the potential. At that scale the structure of the chain follows a random walk, and the distribution of the CG units along the chain is Markovian. This is a general property of the macromolecules [18] when sampled at large enough lengthscales. It should be stressed that in the IECG papers, the analytical potential serves as an approximation, under reasonable assumptions, for the numerical potential that is used in simulations. Having an analytical potential allows one to understand the scaling behavior of the potential with structural parameters, as well as to estimate thermodynamic quantities of interest. The relevant equations are quite sizeable and beyond the scope of this paper, but the complete analytical forms can be found in previous publications [11].

Using this potential we perform simulations of the CG systems, and then compared thermodynamic quantities and structural quantities of interest from these simulations with UA simulation data. The agreement between CG and atomistic descriptions is quantitative, where the direct correlation contribution at large distances, $c(k \rightarrow 0) = c_0$, is the only non-trivial parameter. It is evaluated either from experiments or from theory. Consistency for structural and thermodynamic properties is observed in all comparisons between numerical solutions of the IECG, analytical solutions, UA simulations, and mesoscale simulations [11].

B. The UA↔CG Workflow

The CG representation enables simulations to explore larger chemical systems because it exposes far fewer degrees of freedom than the UA representation. CG can also explore longer timescales because it does not suffer from the geometric constraints within UA systems, such as those caused by entanglements that prohibit efficient dynamics. Unlike bonded monomer chains, CG soft spheres may overlap, which expedites the equilibration of the melt that would have otherwise been entangled. Furthermore, previous work has shown that fundamental thermodynamic properties are fully captured by the CG representation when using our analytically-derived potential [6], [10], [11].

However, after accomplishing equilibration in the CG representation, we still require molecular information on the local scale to account for all properties of interest. By transforming the CG system to a UA representation, we can potentially deliver an equilibrated system having atomistic detail to material scientists at a fraction of the full-atomistic execution time. Furthermore, if we can alternate between the CG representation and the UA representation in an automated manner, then we can simultaneously benefit from the performance of CG and the accuracy of UA. Novel approaches for adaptive resolution in molecular dynamics, in which more interesting regions are coarse-grained at a finer resolution than less interesting regions [19], also require innovative methods for on-the-fly mapping back and forth between UA and CG. Section III-A describes our UA↔CG scientific workflow approach, which accomplishes this feat.

After a CG simulation has equilibrated to a minimal energy configuration, a crucial question is then: which UA system(s) are accurately represented by this arrangement? This raises the notion of the backmapping problem which is the subject of Sections II-C and III-B. Our initial workflow implementation passed data through LAMMPS dump files, which can become a performance bottleneck with large numbers of atoms, saving state often, or transforming between representations often.

C. Backmapping

In homopolymer systems, transforming from the UA representation to the CG representation is straightforward: for each subchain having N_b monomers, the new soft sphere coordinate is simply the center of mass of the subchain. On the other hand, the *reverse* procedure of mapping from the CG representation to the UA representation is not generally well-defined. This transformation of a CG model into a UA model is a popular research topic, commonly referred to as the *backmapping* problem. For our homopolymer system, the backmapping problem is simply stated as follows: given a collection of CG soft sphere chains coordinates, insert monomer chains in such a way that we would recover the original CG configuration if we were to coarse-grain the system again.

It is easy to see that solutions to backmapping problems are not unique, because there are many different UA configurations that could map back to a given CG configuration. Much backmapping work focuses on biomolecules [20], but relatively little work has explored homopolymers. However, efficient backmapping procedures in polymer simulations are imperative for developing a full-fledged adaptively resolved simulations [21].

In previous work [22], we used the Parallelizable Open source Efficient Multibody Software (POEMS) library to avoid backmapping. POEMS treats the internal subchains of each CG soft sphere as a set of *coupled rigid bodies*. This greatly reduces degrees of freedom in the simulation, and has the additional benefit of eliminating the need for solving the backmapping problem. Unfortunately, this approach suffers from poor computational performance, and our performance profiles unequivocally suggest it is due to time spent in POEMS' *Solve*, *initial_integrate*, and *final_integrate* functions. Also, an analysis of internal monomer distances versus endpoint distances (e.g., the monomers that connect adjacent CG spheres) shows that the endpoint bonds stretch to unphysical distances. This issue motivates the need for a backmapping procedure that leads to more physical bond distances throughout the system. Section III-B presents the design of our new backmapping procedure.

III. Design

This section discusses the design of our scientific workflow environment for obtaining high-performance equilibration of polymer melts with atomistic accuracy. We henceforth refer to the overall process as the *UA↔CG workflow*. Section III-A briefly describes the UA↔CG workflow implementation with the goal of providing context and motivation for the backmapping procedure. Further details regarding the workflow implementation can be found elsewhere [22]. Section III-B answers why, when, and how backmapping occurs within the workflow. We then describe our backmapping algorithm, and discuss future directions for possible improvements.

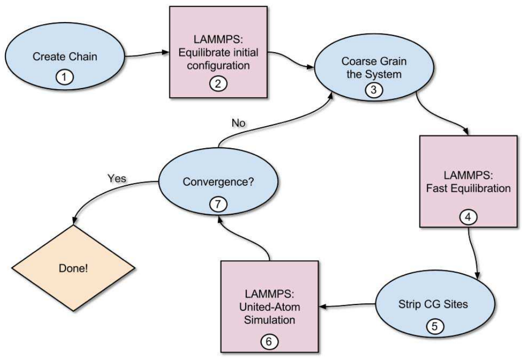
Create Chain
1
LAMMPS: Equilibrate initial configuration
2
Coarse Grain the System
3
No
LAMMPS: Fast Equilibration
4
Yes
Convergence?
7
Done!
LAMMPS: United-Atom Simulation
6
Strip CG Sites
5

Figure 3: The UA↔CG Workflow. Blue circles (1,3,5,7) represent stages of custom programs that either generate coordinates and potentials, transform data for input into LAMMPS, or conduct convergence analysis. Red squares (2,4,6) represent parallel LAMMPS MD simulations executing via MPI. Our workflow system automates this process for a given set of input parameters.

A. The UA↔CG Workflow

The UA↔CG workflow consists of a series of computational programs and analyses that comprise an overall application for quickly stabilizing a randomly generated polymer melt of n chains, each with N monomers per chain. Subsequently, the workflow may be used to do production simulations of the equilibrated polymeric liquid. Figure 3 shows the 7 high-level stages involved in the equilibration workflow, each of which may involve multiple processing steps. Each step is accomplished by one or more programs, applications, or simulations. Before this work, these steps each required manual intervention by a researcher, but now they are automated by our workflow.

The workflow consists of a set of standardized Python wrappers of each Fortran, C, or C++ program that encapsulates Stages 1, 3, 5, and 7 from Figure 3. Some of these programs are actually our own custom versions of LAMMPS tools (such as `chain.f`, which generates random polymer chain systems) that are optimized to run large-scale polymer systems. We use the standard Python argument parsing system, `argparse`, to define all simulation parameters at launch time. We subsequently pass the parameter bundle as a Python object throughout the entire workflow application.

In order for the workflow to be useful, Steps 3, 5, and 7 must have low overhead when compared to the full UA simulation itself. Furthermore, Step 4, which comprises the CG component of the equilibration, must exhibit better performance than UA and must converge towards a physically correct configuration. In our original implementation, the performance of Step 4 was unsatisfactory, especially when considering the vastly fewer degrees of freedom in the CG representation. The next section on backmapping discusses the source of this performance obstacle and our solution. In short, Step 4 executes most efficiently when completely discarding the internal UA coordinates tracked in steps 1-3. Unfortunately, this raises the additional concern of needing to recover those coordinates. This issue is considered in detail in Section III-B.

Much of the data transfer between the workflow steps occurs by processing LAMMPS dump files to create new LAMMPS input files. In most cases, the time spent reading files is negligible compared to the MD simulations, particularly when we configure LAMMPS to write to disk at relatively large timestep intervals. However, the larger the interval, the less information can be included in convergence analysis. In our studies, convergence is detected by examining the percent error change of the radial distribution function [22] within the UA representation. If the average percent error is below a user-defined threshold, then the workflow completes.

B. Backmapping

The database approach for backmapping, in which subchain fragments are stored to a separate repository for insertion into the soft spheres is one alternative. However, this make little sense for polymeric systems, because the necessary size of the database quickly becomes prohibitive. Firstly, we need a separate database of fragments for every value of N_b. Secondly, we need a large number of databases for different monomer types (e.g., different bond distances and masses). Finally, in order to obtain good statistics, many different suitable configurations are required for each possible N_b, and monomer type.

The approximate reconstruction approach for backmapping is far more suitable for homopolymer systems. We do not require a perfect solution, because a quick UA simulation can remove any geometric strain leaving us with an equilibrated system (Step 6 of the workflow from Figure 3). In our first implementation, we take advantage of this technique by generating a very simple configuration of chains on a regular Cartesian grid. The steps for constructing a polymer melt on such a grid are as follows:

1) Store the center of mass coordinates for each soft sphere along the polymers.
2) Calculate the midpoints between the center of mass coordinates of each pair of adjacent spheres.
3) Initialize a regular grid across the simulation box with grid point distances equal to the desired bond length.
4) Redefine the above midpoints as the desired *endpoints* of each subchain and place onto the nearest grid point.
5) Generate paths connecting each pair of endpoints with the “Manhattan distance” length between endpoints.
6) Randomly extend each the path to the desired number of edges, N_b, by inserting *extensions*.

Figure 4 illustrates a simple example of the random extension in step 6. The leftmost graph is the result of steps 1-5 for a single subchain. By removing edge $\mathtt{e}_1$ and inserting the $\mathtt{ext}_1$ extender in its place, we extend the length of the chain's path by 2. Next, edge $\mathtt{e}_2$ is randomly selected and the path is extended by another 2 bond lengths. If we desire 10 monomers per sphere, then the algorithm is complete. Figure 5 shows a full example with roughly 30 monomers per soft sphere. For brevity, we omit certain implementation details in this algorithm description. The fully commented source code can be found in the UA↔CG Workflow repository, freely available online [23].

This backmapping approach has the advantage of being straightforward to implement, lightweight, relatively general, and potentially parallel. For polyethylene, however, this approach has the disadvantage of producing chains with unphysical bond angles along the carbon backbone, which means that a longer simulation is required to equilibrate the newly generated UA system. An extension to the regular grid approach is to instead construct a *tetrahedral* grid, in which angles between grid points are forced to be $109.5°$ instead of $90°$. We leave the tetrahedral implementation and comparing its equilibration requirements as future work.

IV. Experimental

Stages 1, 3, and 5 of the workflow only transform a *single* simulation snapshot. The algorithms are $O(m)$ (m is the total number of monomers), involve no communication, and typically comprise far less than 1% of the total workflow execution time. Stages 2, 4, and 6 potentially involve millions or billions of snapshots with much communication as particles advance. Stage 4 alone may consume over 90% of the workflow execution time (depending on how many time steps are assigned to each stage), making it the clear performance bottleneck. Therefore, this section focuses on strong and weak scaling experiments of Stage 4 in the workflow, emphasizing the benefit of using our analytically-derived multiblock potential in the CG representation.

A. Experimental Setup and Evaluation

Experiments were conducted on the ACISS cluster located at the University of Oregon. We use the 128 generic compute nodes, each an HP ProLiant SL390 G7 with 12 processor cores per node (2x Intel X5650 2.67 GHz 6-core CPUs) and 72 GB of memory per node. ACISS employs a 10 gigabit HP Voltaire 8500 Ethernet switch that connects all compute nodes and storage fabric. The operating system is RedHat Enterprise Linux 6.2, and we used Intel version 14.0 compilers with OpenMPI 1.7. The latest version of LAMMPS (version 10-Feb-2015) was used with a slight modification to the chain generation tool to enable massive particle scaling.

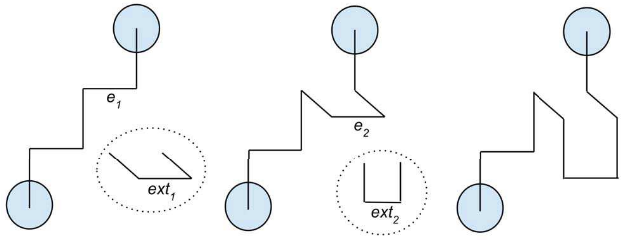

Figure 4: Simple example of regular grid backmapping.

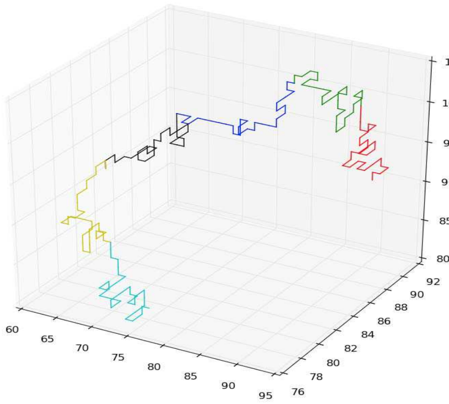

Figure 5: A randomly generated path with fixed endpoints on a regular grid. This is the UA configuration backmapped from a CG representation with ~30 monomers per sphere. Each color corresponds to a different subchain, and units are in angstroms.

We also include scaling experiments conducted on the Hopper system at NERSC. Hopper is a Cray XE6 cluster, where each compute node has 2 AMD 12-core MagnyCours (24 cores total) running at 2.1 GHz. There are 32 GB of DDR3 RAM per node. We use the default LAMMPS 20140628 module, and the default PGI Cray compiler, version 14.2-0.

B. UA versus CG Performance

Figures 6 and 7 show the strong and weak scaling of the UA versus CG components of the workflow on ACISS. These timings measure 500 femtoseconds of simulation time, then extrapolate to hours of execution time per nanosecond of simulation time, since we know that several nanoseconds are typically required to reach equilibration. The auxiliary workflow programs (CGgen, UAInit, and BMinit) never take more than a few seconds, which constitutes far less than 0.1% of the overall runtime for full simulations, so we do not include those times in these measurements.

Figure 6 shows a strong scaling experiment on ACISS, where each data point corresponds to a simulation that *represents* 200,000 UA monomer sites. In the UA case, we run 20 chains, each with 10,000 monomers. The number of sites in the CG representation, however, depends on the granularity of the decomposition. For instance, with 10 sites per chain, each soft sphere contains 1,000 monomers, and only 200 total soft spheres are simulated. Therefore, it comes at no great surprise that the CG component of the workflow stops strong scaling after 8 nodes on ACISS (or 96 processes). At 8 nodes, only about 2 spheres reside in

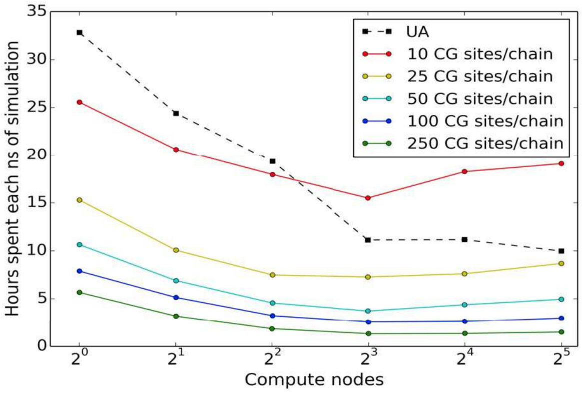

Figure 6: Strong scaling on ACISS for 20 chains each with 10,000 monomers. As expected, CG does not strong scale to as many processes as UA because it simulates fewer particles. However, absolute performance is usually far better for CG than UA, even at scale. Including more sites per chain is faster *per timestep*, but this ignores the rate of convergence of thermodynamic quantities.

each process, and at 16 nodes, there are more processes than spheres, which is highly undesirable for an application like LAMMPS where communication between neighboring atoms plays a large role. At 250 sites per chain, there are a total of 5,000 soft spheres, which is still far too small to exhibit good strong scaling at large numbers of processes. It may be unexpected that the absolute performance is better for larger numbers of sites per chain, and is best at 250. While this may seem counter-intuitive, the phenomenon is due to the fact that lower numbers of sites per chain require longer-ranged interactions. The larger cutoff results in more neighbor particles per process, and the application becomes communication bound. Also, these measurements do not take into consideration the rate of convergence of thermodynamic quantities, which we expect to be faster for coarser models.

Figure 7 shows the corresponding weak scaling experiment on ACISS, in which the number of chains per compute node is held constant at 20 chains for each execution. For instance, at 32 nodes, UA simulates 640 chains for a total of 6.4 million particles. CG also simulates 640 chains at 32 nodes, but at 200 sites per chain 32,000 spheres are tracked. The plot clearly shows that tracking 32,000 soft spheres exhibits far better weak scaling than simulating 6.4 million atoms. This may not be so surprising, but the important thing to note is that using our potential in the CG simulation can accurately produce an equilibrated system with the same thermodynamic properties as the UA simulation. Previous work has demonstrated almost perfect recovery of pressure, free energy, and structural correlations at several different granularities of up to 1,000 soft spheres per chain [11]. Until this work, simulating 10,000 to 1 million soft spheres per chain was impractical.

To provide evidence that the above results are independent of the cluster architecture, we ran a comparable experiment on the Hopper system at NERSC. Figure 8 shows a strong scaling experiment with 20 chains and 10,000 monomers per chain for UA versus CG with 200 sites per chain. As before, the CG simulation stops strong scaling before UA, likely due to the vastly smaller number of particles tracked (1,000 versus 200,000). On the other hand, Figure 9 confirms that the CG simulation weak scales far better than UA, up to 3072 processes.

We also ran equivalent weak scaling experiments with 2 chains per node and 100,000 monomers per chain with almost identical performance results. To avoid redundancy, we omit those results here, but the fact that this is possible is exciting for future validation work and our ability to simulate systems that reflect real-world polyethylene melts.

While it is possible to achieve speedups on the order of 10x for CG simulations based on IECG theory, it is important to converge to a useful configuration in UA space. Measurements of our backmapping procedure are on the order of a few seconds, even for systems containing millions of particles. Compared to the several hours required to complete a full equilibration, this time is negligible, and gives more significance to our CG versus UA comparisons.

V. Related Work

Most related CG models (such as within the MARTINI force field [15]) use numerically optimized potentials, with the problem that numerical errors in the optimization of the potential can propagate to other physical quantities, such as thermodynamic properties. In general, numerically optimized CG potentials have problems with transferability and correct representability. It has been observed that numerical CG potentials that are optimized to reproduce some physical quantity correctly, such as the structure, may not correctly reproduce other quantities, such as the pressure or the free energy of the system. This problem is not present in our analytically-based CG simulations,

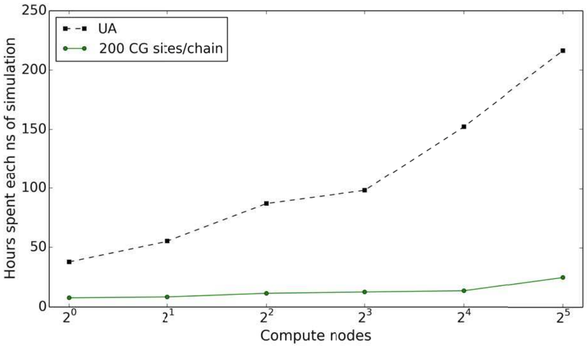

Figure 7: Weak scaling on ACISS for 20 chains per compute node, each chain containing 10,000 monomers

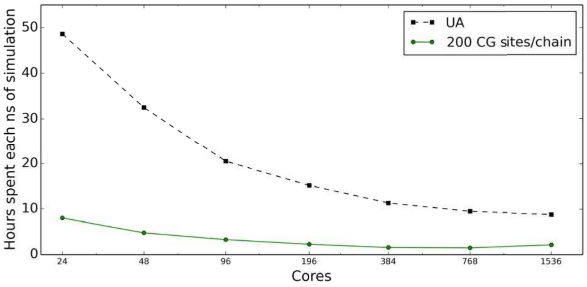

Figure 8: Strong scaling on Hopper for 20 chains each with 10,000 monomers.

because the formal solution of the potential, as well as of the structural and thermodynamic quantities of interest, ensures their consistency across variable levels of resolution if the CG description.

Many others have studied the backmapping problem, and due to the non-unique nature of the solutions, many different methods have been proposed such as fragment-based libraries [24] and optimization procedures with gradually relaxing harmonic constraints between the UA and CG systems [25]. Furthermore, particular backmapping procedures tend to be specific to the problem under consideration, often by necessity. For example, backmapping polymer models have been applied to nonequilibrium shear flows by Chen et al. [26]. For CG models of polymers with rigid side groups, Ghanbari et al. use simple geometric properties of the molecular fragments to reconstruct the atomic system [27]. Our implementation in Section III-B supports homopolymers with a relatively constant bond distance (although, it would certainly be possible to generalize to heteropolymers or block copolymers in future implementations). The general sentiment in the CG research community is that backmapping procedures are important for supporting adaptive resolution capabilities, but we desire a general approach for backmapping. We believe that online construction of realistic configurations offers the most promising approach for a general and high performance backmapping procedure.

A plethora of related research and development focuses on optimizing the performance and productivity of scientific workflows. One of the most popular frameworks is Kepler, which has recently presented full-fledged workflows for drug design, and developments towards supporting popular distributed data-parallel patterns, such as MapReduce [28]. Other scientific workflow environments include Pegasus [29], and Taverna [30]. Instead of committing to a full-blown workflow framework up-front, we tap into the advantages of using a considerably lightweight system for managing input parameters while benefiting from simple data provenance and template processing. For instance, our workflow is trivial to setup on any Linux based machine with Python and LAMMPS installed.

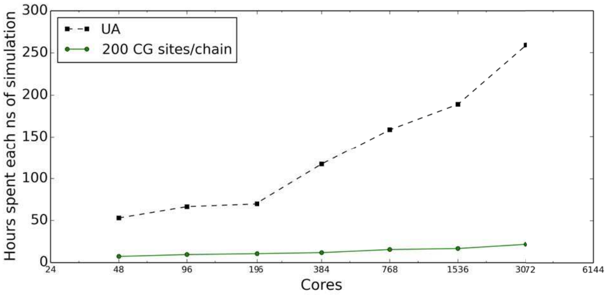

Figure 9: Weak scaling on Hopper. Each UA simulation runs 20 chains per node with 10,000 monomers per chain. The CG simulations have a granularity of 200 soft sphere sites per chain, so each sphere contains 50 monomers. Our CG simulations weak scales nicely, and offer a speedup of about 12x at 3,072 cores.

VI. Conclusion

Coarse-graining methods benefit from reduced computational requirements, allowing us to simulate systems with a realistic number of atoms relative to laboratory bulk experiments. Few polymer simulation studies have explored long-chain configurations, despite their importance for studying real-world systems. This paper has presented a customized set of software tools for running such large systems with LAMMPS as the primary molecular dynamics framework. We use the analytically based IE-CG potential, which has been shown to preserve both thermodynamic quantities and bulk properties, unlike other numerically based potentials. In addition to our collection of individual software tools, we have implemented a full-fledged scientific workflow that enables automatic transformation between UA and CG representations. The CG to UA backmapping problem is handled by randomly generating a polymer onto a regular Cartesian grid followed by a short UA equilibration.

The UA$\leftrightarrow$CG workflow enables 4 noteworthy features for conducting large-scale polymer studies:

1) the automated setup of CG systems based on a corresponding UA configuration
2) excellent parallel performance when applying IECG theory
3) a backmapping procedure from the CG to the UA representation, restoring local molecular information
4) the potential to iterate through several cycles of the workflow loop

By having low-overhead tools between workflow phases, we can focus on the performance comparison of CG versus UA. Not only do our scaling experiment results show a general benefit of using our CG potential over straightforward UA, they also suggest the effectiveness of our new workflow for transforming between UA and CG. We have quantified what performance improvements to expect when in the CG component of the workflow, and we better understand how to choose the granularity factor to best exploit this benefit.

Now that simulation of long-chain polymer systems is possible, efficient, and dynamically transformable into either UA or CG representation, future work will formally validate the thermodynamic quantities similarly to what has been done for $\leq$1000 monomers per chain. By quickly switching back and forth from UA to CG, we open doors to new studies of polymeric systems while maintaining simulation accuracy and efficiency.

ACKNOWLEDGMENTS

D. Ozog is supported by the U.S. Department of Energy (DOE) Computational Science Graduate Fellowship program under contract DE-FG02-97ER25308. The research at the University of Oregon is supported by grants from DOE, Office of Science, under contracts DE-SC0006723, DE-SC0012381, and DE-SC0005360; and the National Science Foundation under award CHE-1362500.

REFERENCES

[1] S. K. Kumar, N. Jouault, B. Benicewicz, and T. Neely, "Nanocomposites with Polymer Grafted Nanoparticles," *Macromolecules*, vol. 46, no. 9, pp. 3199–3214, 2013.

[2] A. Smessaert and J. Rottler, "Recovery of polymer glasses from mechanical perturbation," *Macromolecules*, vol. 45, no. 6, pp. 2928–2935, 2012.

[3] J. Baschnagel, K. Binder, P. Doruker *et al.*, "Bridging the Gap Between Atomistic and Coarse-Grained Models of Polymers: Status and Perspectives," in *Viscoelasticity, Atomistic Models, Statistical Chemistry*, 2000, vol. 152, pp. 41–156.

[4] A. Peacock, *Handbook of Polyethylene: Structures, Properties, and Applications*. CRC Press, 2000.

[5] E. J. Sambriski and M. G. Guenza, "Theoretical coarse-graining approach to bridge length scales in diblock copolymer liquids," *Phys. Rev. E*, vol. 76, p. 051801, Nov 2007.

[6] A. J. Clark and M. G. Guenza, "Mapping of Polymer Melts onto Liquids of Soft-Colloidal Chains," *The Journal of Chemical Physics*, vol. 132, no. 4, 2010.

[7] G. Yatsenko, E. J. Sambriski, M. A. Nemirovskaya, and M. Guenza, "Analytical soft-core potentials for macromolecular fluids and mixtures," *Phys. Rev. Lett.*, vol. 93, p. 257803, Dec 2004.

[8] I. Y. Lyubimov and M. G. Guenza, "Theoretical Reconstruction of Realistic Dynamics of Highly Coarse-Grained cis-1,4-polybutadiene Melts," vol. 138, no. 12, p. 120000, Mar. 2013.

[9] J. McCarty and M. Guenza, "Multiscale modeling of binary polymer mixtures: Scale bridging in the athermal and thermal regime," *The Journal of Chemical Physics*, vol. 133, no. 9, p. 094904, 2010.

[10] A. J. Clark, J. McCarty, and M. G. Guenza, "Effective Potentials for Representing Polymers in Melts as Chains of Interacting Soft Particles," *The Journal of Chemical Physics*, vol. 139, no. 12, 2013.

[11] J. McCarty, A. J. Clark, J. Copperman, and M. G. Guenza, "An analytical coarse-graining method which preserves the free energy, structural correlations, and thermodynamic state of polymer melts from the atomistic to the mesoscale," *The Journal of Chemical Physics*, vol. 140, no. 20, 2014.

[12] A. Clark, J. McCarty, I. Lyubimov, and M. Guenza, "Thermodynamic Consistency in Variable-Level Coarse Graining of Polymeric Liquids," *Phys. Rev. Lett.*, vol. 109, p. 168301, Oct 2012.

[13] J. F. Dama, A. V. Sinitskiy, M. McCullagh, J. Weare, B. Roux, A. R. Dinner, and G. A. Voth, "The theory of ultra-coarse-graining. 1. general principles," *Journal of Chemical Theory and Computation*, vol. 9, no. 5, pp. 2466–2480, 2013.

[14] D. Reith, M. Pütz, and F. Müller-Plathe, "Deriving effective mesoscale potentials from atomistic simulations," *Journal of Computational Chemistry*, vol. 24, no. 13, pp. 1624–1636, 2003.

[15] S. J. Marrink and D. P. Tieleman, "Perspective on the martini model," *Chem. Soc. Rev.*, vol. 42, pp. 6801–6822, 2013.

[16] J. Hansen and I. R. McDonald, *Theory of Simple Liquids*. Academic Press, London, 1991.

[17] K. S. Schweizer and J. G. Curro, "Integral equation theories of the structure, thermodynamics, and phase transitions of polymer fluids," *Advances in Chemical Physics, Volume 98*, pp. 1–142, 1997.

[18] P.-G. De Gennes, *Scaling concepts in polymer physics*. Cornell university press, 1979.

[19] M. Praprotnik, L. Delle Site, and K. Kremer, "Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly," *The Journal of Chemical Physics*, vol. 123, no. 22, 2005.

[20] T. A. Wassenaar, K. Pluhackova, R. A. Böckmann, S. J. Marrink, and D. P. Tieleman, "Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models," *Journal of Chemical Theory and Computation*, vol. 10, no. 2, pp. 676–690, 2014.

[21] C. Peter and K. Kremer, "Multiscale simulation of soft matter systems - from the atomistic to the coarse-grained level and back," *Soft Matter*, vol. 5, pp. 4357–4366, 2009.

[22] D. Ozog, J. McCarty, G. Gossett, A. D. Malony, and M. Guenza, "Fast equilibration of coarse-grained polymeric liquids," *Journal of Computational Science*, vol. 9, no. 0, pp. 33 – 38, 2015.

[23] "Fast-Equilibration-Workflow source code repository," 2015.

[24] P. J. Stansfeld and M. S. Sansom, "From Coarse Grained to Atomistic: A Serial Multiscale Approach to Membrane Protein Simulations," *Journal of Chemical Theory and Computation*, vol. 7, no. 4, pp. 1157–1166, 2011.

[25] A. J. Rzepiela, L. V. Schäfer, N. Goga, H. J. Risselada, A. H. De Vries, and S. J. Marrink, "Reconstruction of atomistic details from coarse-grained structures," *Journal of Computational Chemistry*, vol. 31, no. 6, pp. 1333–1343, 2010.

[26] X. Chen, P. Carbone, G. Santangelo, A. Di Matteo, G. Milano, and F. Muller-Plathe, "Backmapping coarse-grained polymer models under sheared nonequilibrium conditions," *Phys. Chem. Chem. Phys.*, vol. 11, pp. 1977–1988, 2009.

[27] A. Ghanbari, M. C. Böhm, and F. Müller-Plathe, "A Simple Reverse Mapping Procedure for Coarse-Grained Polymer Models with Rigid Side Groups," *Macromolecules*, vol. 44, no. 13, pp. 5520–5526, 2011.

[28] J. Wang, D. Crawl, and I. Altintas, "A Framework for Distributed Data-Parallel Execution in the Kepler Scientific Workflow System," *2012 International Conference on Computational Science (ICCS 2012)*, vol. 9, no. 0, pp. 1620 – 1629, 2012.

[29] E. Deelman, G. Singh, M.-H. Su *et al.*, "Pegasus: A Framework for Mapping Complex Scientific Workflows Onto Distributed Systems," *Scientific Programming*, vol. 13, no. 3, pp. 219–237, Jul. 2005.

[30] T. Oinn, M. Addis, J. Ferris *et al.*, "Taverna: a Tool for the Composition and Enactment of Bioinformatics Workflows," *Bioinformatics*, vol. 20, pp. 3045–3054, 2004.