Convergence of data generation and analysis in the biomolecular simulation community

Oliver Beckstein[1], Geoffrey Fox[2], Shantenu Jha[3,4]

[1]Arizona State University, Tempe AZ <obeckste@asu.edu>; [2]Indiana University, Bloomington IN <gcf@indiana.edu>; [3]Rutgers University, Piscataway NJ; [4]Brookhaven National Laboratory <shantenu.jha@rutgers.edu>

The changing nature of biomolecular simulations

In the biomolecular simulation (BMS) community, classical molecular dynamics (MD) simulations enable the elucidation of the relationship between the structure of biomolecules such as proteins, nucleic acids, or lipids and their function via their dynamics. MD simulations account for approximately one quarter of the service units used on XSEDE resources. Although traditionally the generation of the data has been the computational bottleneck and has been highly optimized, more and more the analysis of the data is becoming a rate limiting step. Within the NSF DIBBs SPIDAL project we have been working on leveraging HPC resources for the analysis of BMS data [1], starting from two widely adopted software packages in the community, cpptraj [2] and MDAnalysis [3,4].

Current state of the art simulations are performed at the atomic level and include the niomolecules and their environment such as water, ions, lipids, and small molecules. Typical system sizes range from $O(10^3)$ to $O(10^6)$ atoms with some exceptionally large systems up to $\sim 10^8$ [5]. Simulations integrate the equations of motion of all atoms using femtosecond timesteps. The positions (and possibly velocities) of the atoms are saved to a trajectory file at regular intervals, typically every 1 to 100 ps. Current simulation lengths typically achieve ≤ 10 µs although on special hardware up to 1 ms has been achieved for small systems with $O(10^4)$ atoms [6], while massively distributed simulations can produce aggregate data of up to 6 ms [7]. Advances in hardware (GPUs, FPGA/custom hardware, exascale resources such as Summit) and software (e.g. GPU-optimized codes) lead to a steady increase in the trajectory sizes [8], currently in the hundreds of GB to a few TB range.

A common approach is to run a single or a few repeats of simulations for a fixed condition with the aim to capture equilibrium behavior. Long continuous trajectories have provided valuable insights in protein folding in a unbiased manner [6]. However, increasingly the emphasis is on sampling of rare events and quantitative predictions of free energies and rates, which necessitates enhanced sampling approaches [9, 10, 11] that run ensembles of tens to hundreds of coupled simulations. Exascale computing promises to make such calculations much more feasible and more widespread. The trajectories from enhanced sampling runs have to be analyzed as a single dataset; the size of the datasets (approaching hundreds of TB [12]) will make it infeasible to move the data away from the HPC system where they were produced and their analysis will take too long with current serial approaches. The challenge becomes to design computational environments that support both data generation and analysis efficiently and to develop analysis software that makes best use of resources that have been geared towards data production.

From offline to online analysis

Thus, concomitant with increased computing capabilities is the opportunity and the need for sophisticated and efficient analysis of unprecedented volumes of data generated from simulations. The temporal coupling of Molecular Dynamics simulations generating data (producer) to analysis of the data (consumer) can be classified into three broad categories,

I. *Data Reduction*: This is classic scenario, where in-situ (real-time) analysis of data is performed to reduce the volume of data that needs to eventually be stored or output to disc. Original drivers of data reduction were poor file system performance, but recent advances in the ability to "compute only what you need" [13] scenarios has resulted in several approaches to analysing data once and only once.

2. *Streaming Data into Analysis*: There have been advances in stream-based algorithms of traditional analysis algorithms which benefit from incremental data availability, thus necessitating the ability of large volumes of data to be streamed directly from simulations to analysis. The need to stream data directly into simulations is not confined to stream-based analysis algorithms; several online learning algorithms [14] benefit from increased and incremental data.

3. *Adaptive Simulations*: Arguably the coupled simulation-analysis scenario that has received the greatest attention thus far, is the general class of algorithms referred to as adaptive algorithms, and in particular adaptive ensembles simulations [15]. In adaptive algorithms the intermediate data generated by simulations is used to guide the evolution of the next stage of simulations. Traditional examples of these include Markov State Model (MSM) and variants thereof, but recently more sophisticated ML-driven approaches to steering simulations (ML-driven-MD) have been both proposed and implemented [11]. The motivation for adaptive simulations varies from "better, faster and greater" sampling of a very large phase space [15], to the efficient utilization of limited computing resources [16]. Ref [17] discusses a software system that supports multiple adaptive algorithms that significantly increase simulations efficiency.

AI-driven analysis and simulation

ML is being used to analyze the results of molecular dynamics simulations (e.g., binding affinities [12], folding [14], phase diagrams [20], or Tang's contribution to this meeting predicting stability). An exciting idea is to use AI-driven analysis to advance MD simulations or whole ensembles of simulations based on the phase space that has already been sampled. Such AI-driven autotuning has been shown to be able to increase time steps in QM MD [19] and suggest parameters (such as timestep size, spatial meshes, internal polarization densities) to be used in the simulation [21] but more widespread application of these ideas will likely require meeting the challenges of converging simulations and analysis as outlined above.

References

[1] I. Paraskevakos, A. Luckow, M. Khoshlessan, G. Chantzialexiou, T. E. Cheatham, O. Beckstein, G. Fox, and S. Jha. Task-parallel analysis of molecular dynamics trajectories. In ICPP 2018: 47th International Conference on Parallel Processing, August 13–16, 2018,

Eugene, OR, USA, New York, NY, USA, August 13–16 2018. Association for Computing Machinery, ACM.

[2] D. R. Roe and T. E. Cheatham. PTRAJ and CPPTRAJ: Software for processing and analysis of molecular dynamics trajectory data. Journal of Chemical Theory and Computation, 9(7):3084–3095, 2013.

[3] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. J Comp Chem, 32:2319–2327, 2011.

[4] R. J. Gowers, M. Linke, J. Barnoud, T. J. E. Reddy, M. N. Melo, S. L. Seyler, D. L. Dotson, J. Domanski, S. Buchoux, I. M. Kenney, and O. Beckstein. MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations. In S. Benthall and S. Rostrup, editors, Proceedings of the 15th Python in Science Conference, pages 98–105, Austin, TX, 2016. SciPy.

[5] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. Nature, 497(7451):643–6, May 2013.

[6] D. E. Shaw, R. O. Dror, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers, E. Chow, M. P. Eastwood, D. J. Ierardi, J. L. Klepeis, J. S. Kuskin, R. H. Larson, K. Lindorff-Larsen, P. Maragakis, M. A. Moraes, S. Piana, Y. Shan, and B. Towles. Millisecond-scale molecular dynamics simulations on anton. In SC ’09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, pages 1–11, New York, NY, USA, 2009. ACM.

[7] S. Chen, R. P. Wiewiora, F. Meng, N. Babault, A. Ma, W. Yu, K. Qian, H. Hu, H. Zou, J. Wang, S. Fan, G. Blum, F. Pittella-Silva, K. A. Beauchamp, W. Tempel, H. Jiang, K. Chen, R. Skene, Y. G. Zheng, P. J. Brown, J. Jin, C. Luo, J. D. Chodera, and M. Luo. The dynamic conformational landscapes of the protein methyltransferase SETD8. bioRxiv, 2018. DOI: 10.1101/438994

[8] T. Cheatham and D. Roe. The impact of heterogeneous computing on workflows for biomolecular simulation and analysis. Computing in Science Engineering, 17(2):30–39, 2015.

[9] T. Maximova, R. Moffatt, B. Ma, R. Nussinov, and A. Shehu. Principles and overview of sampling methods for modeling macromolecular structure and dynamics. PLoS Comput Biol, 12(4):1–70, 04 2016.

[10] M. C. Zwier and L. T. Chong. Reaching biological timescales with all-atom molecular dynamics simulations. Curr Opin Pharmacol, 10(6):745–52, Dec 2010.

[11] J. D. Chodera and F. Noé. Markov state models of biomolecular conformational dynamics. Current Opinion in Structural Biology, 25:135 – 144, 2014. Theory and simulation / Macromolecular machines.

[12] A. Pérez, G. Martínez-Rosell, and G. D. Fabritiis. Simulations meet machine learning in structural biology. Current Opinion in Structural Biology, 49:139 – 144, 2018.

[13] I. Foster, M. Ainsworth, B. Allen, J. Bessac, F. Cappello, J. Y. Choi, E. Constantinescu, P. E. Davis, S. Di, W. Di, H. Guo, S. Klasky, K. K. Van Dam, T. Kurc, Q. Liu, A. Malik, K. Mehta, K.

Mueller, T. Munson, G. Ostouchov, M. Parashar, T. Peterka, L. Pouchard, D. Tao, O. Tugluk, S. Wild, M. Wolf, J. M. Wozniak, W. Xu, and S. Yoo. Computing just what you need: Online data analysis and reduction at extreme scales. In F. F. Rivera, T. F. Pena, and J. C. Cabaleiro, editors, Euro-Par 2017: Parallel Processing, pages 3–19, Cham, 2017. Springer International Publishing.

[14] D. Bhowmik, M. T. Young, S. Gao, and A. Ramanathan. Deep clustering of protein folding simulations. bioRxiv, 2018. Doi: 10.1101/339879

[15] Peter M. Kasson and Shantenu Jha. Adaptive ensemble simulations of biomolecules. Current Opinion in Structural Biology 2018. 52:87-94. DOI:10.1016/j.sbi.2018.09.005

[16] Concurrent and Adaptive Extreme Scale Binding Free Energy Calculations. Jumana Dakka, Kristof Farkas-Pall, Matteo Turilli, David W Wright, Peter V Coveney, Shantenu Jha, published in IEEE eScience 2018 (arXiv 1801.01174)

[17] Adaptive Ensemble Biomolecular Simulations at Scale. Vivek Balasubramanian, Travis Jensen, Matteo Turilli, Peter Kasson, Michael Shirts, Shantenu Jha. 2018 (arXiv 1804.04736)

[19] V. Botu and R. Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. International Journal of Quantum Chemistry, 115(16):1074–1083, 2014.

[20] M. Spellings and S. C. Glotzer. Machine learning for crystal identification and discovery. AIChE Journal, 64(6):2198–2206, 2018.

[21] JCS Kadupitiya, Geoffrey C. Fox and Vikram Jadhao, “Machine Learning for Parameter Auto-tuning in Molecular Dynamics Simulations: Efficient Dynamics of Ions near Polarizable Nanoparticles”, paper in preparation