Towards General-Purpose Long-Timescale Molecular Dynamics Simulation on Exascale Supercomputers with Data Processing Units

Matea Turalija*, Milan Petrović† and Božidar Kovačić‡
*Group for Applications and Services on Exascale Research Infrastructure (GASERI),
†Laboratory for Semantic Technologies (SemTech) and
‡Laboratory for Information Systems Development (InfoSysDev)
Faculty of Informatics and Digital Technology, University of Rijeka, Rijeka, Croatia
{matea.turalija,milan.petrovic,bkovacic}@inf.uniri.hr

***Abstract*—Molecular dynamics (MD) simulation provides the atomic-level characterization of biomolecular systems and their transitions, such as conformational changes in proteins. The computational demands of such simulations and limits of parallelization techniques have prevented simulations of real-world systems from reaching the microsecond timescales, which are relevant for real-world applications. The notable exceptions are the supercomputers specifically designed for MD simulations. An example of such supercomputers is the Anton supercomputer, nowadays in its third iteration, which uses a substantial number of application-specific integrated circuits (ASICs) for MD simulation and is not generally available. Recent advances in algorithms, software, and hardware towards exascale supercomputing have made microsecond-timescale simulations of practically relevant biomolecular systems reachable within days. Data processing units (DPUs) are already being used in data centers for the in-flight processing of network packets (e.g. encryption, decryption, and intrusion detection) and are expected to be used in future exascale supercomputers in some form. The usage of DPUs in the supercomputers unlocks the potential to accelerate MD simulations that were previously available only in networking ASICs in supercomputers such as Anton. This paper proposes the usage of DPUs for MD simulation acceleration in an innovative way inspired by the Anton supercomputer.**

***Index Terms*—molecular dynamics, NVIDIA, data processing unit, DPU, heterogeneous computing, BlueField, DOCA, Anton, Desmond, GROMACS, LAMMPS**

I. Introduction

Molecular dynamics (MD) simulations allow scientists to study the behavior of biological macromolecules and are becoming a more important tool in the field of molecular biology [1]. The past decades have seen breakthroughs in MD performances in the form of parallelization, algorithmic advances, and hardware specialization, leading to several scientific discoveries [2].

Exploiting parallelism to accelerate a computation typically involves dividing it into many small tasks that can be assigned to different processing elements, which may require significant communication between nodes [3]. Performing MD simulations at the scale and speed necessary to address research questions requiring microsecond simulation timescales within practical time frames, however, remains an extreme challenge in high-performance computing and requires specialized hardware [4, 5].

The advent of NVIDIA's Compute Unified Device Architecture (CUDA) technology in the late 2000s enabled general computing on graphics processing units (GPUs) [6]. Specifically, it enabled the acceleration of scientific and professional software by offloading large parts of the computation from the CPU to the GPU both in the context of workstations and high-performance computers. NVIDIA's catalog of GPU accelerated applications [7] today contains hundreds of readily available applications, including MD simulation applications such as GROMACS [8, 9]. After the GPUs, especially with the advent of artificial neural networks and machine learning demand for computing power, a number of hardware solutions similar to GPUs appeared; somewhat general field-programmable gate arrays (FPGAs) were followed by domain-specific application-specific integrated circuits (ASICs) such as Google tensor processing units (TPUs) [10].

Several state-of-the-art MD simulation software packages use parallel computing technologies to utilize available hardware resources, including both offload to accelerators, often using CUDA, and distributed computing technologies using Message Passing Interface (MPI). The examples of such software packages, listen in alphabetical order, are Amber [11], CHARMM [12], GROMACS [13], HOOMD-blue [14], LAMMPS [15], OpenMM [16], and Tinker [17].

Amber is a set of numerous programs that work together to set up, run, and perform analysis of MD simulations. It supports both CUDA and MPI. While it is licensed under a proprietary license, its tools are licensed under GNU General Public License (GPL).

CHARMM is a general and flexible molecular simulation and modeling program that uses classical and quantum mechanical energy functions for simulation of various molecular systems. CHARMM supports MPI and also supports CUDA via OpenMM (described below). It is licensed under a proprietary license.

GROMACS is one of the most widely used open-source and free software codes in chemistry, used primarily for

dynamical simulations of biomolecules and provides a rich set of calculation types, preparation, and analysis tools. It supports both CUDA and MPI. It is licensed under the GNU Lesser General Public License (LGPL).

HOOMD-blue is a particle simulation toolkit for molecular dynamics and hard particle Monte Carlo simulations. It supports both CUDA and MPI. It is licensed under 3-Clause BSD License.

LAMMPS (an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator) is a classical molecular dynamics code with a focus on materials modeling, which offers potentials for metals, biomolecules, and others. It supports both CUDA and MPI. It is licensed under GNU GPL.

OpenMM is an MD simulation toolkit focused on extensibility. Its users can to easily add forces with novel functional forms, new integration algorithms, and new molecular simulation protocols, among others. It supports CUDA, but not MPI due to its focus on single-node computer systems with accelerators and usage of Markov state models for multi-node simulations [18]. It is licensed under the permissive MIT and LGPL licenses.

Tinker is a complete and general package for molecular mechanics and dynamics simulation. It supports both CUDA and MPI It is licensed under a proprietary license.

We observe that all of the state-of-the-art molecular dynamics support CUDA and most support MPI, with the sole exception of OpenMM for which the lack of support is a choice. Therefore, when targeting new hardware for running MD simulation, it would be easier to enable wide usage if it would be possible to support it via CUDA or MPI.

Nowadays, in addition to CPUs and GPUs, there is a third type of processing unit that is completely new, the data processing unit (DPU). In the computer system including the CPU, the GPU, and the DPU, the computation is ideally divided as follows:

- CPU is the core of the system that does general-purpose computation,
- the GPU accelerates parallel computation (such as single instruction, multiple data (SIMD) operations), and
- the new DPU is designed to process data that gets sent and received over the network.

To be able to process in-flight network data, the DPU has to contain its own CPU- or GPU-like accelerator in addition to the network interface. In this way the network interface gets augmented with additional computational functionality and some vendors are calling these devices smart network interface controllers (SmartNICs). There are several DPUs/SmartNICs available on the market at present:

- Pensando (acquired by AMD) Capri, Elba, and DSC,
- Broadcom Stingray,
- Fungible F1 and S1,
- Intel Infrastructure Processing Unit (IPU) C5000X-PL Platform and Platform Codenamed Oak Springs Canyon,
- Kalray K200-LP,
- Marvell Technology OCTEON and ARMADA,
- Mellanox Technologies (acquired by NVIDIA) BlueField, ConnectX, and Innova.

In addition to the broad interest from processor and networking hardware vendors in producing such devices, the acquisitions of Pensando and Mellanox by AMD and NVIDIA (respectively) show the expectations by hardware vendors of increasing application support as well as growth of usage in data centers and supercomputers.

In order to reach exascale supercomputing at the same power envelope as petascale supercomputing, a large increase in performance per watt is needed. GPUs, FPGAs, and ASICs have demonstrated the potential of heterogeneous architecture in the achievement of a greater density of computing elements. A possible approach includes adding compute elements to other components, including network and storage [19, 20], where existing commercially available DPUs are an option. When equipped with DPUs, conventional general-purpose supercomputers are closer to specialized supercomputers for MD simulations equipped with custom networking ASICs such as Anton [4] and MDGRAPE [5].

We decided to focus on using NVIDIA BlueField DPUs due to hardware being available to us for academic research and having a well-documented application programming interface (API), but we expect the DPUs from other vendors with comparable features to be applicable as well. NVIDIA's line of BlueField data processing units (DPUs) is designed with the goal of adding compute elements to network interface controllers (NICs), specifically ARM CPU and an optional programmable accelerator that can perform advanced packet processing. For example, Netflix is using SmartNICs [21] for the TLS encryption of HTTP(S) traffic, offloading the encryption from the central processing units (CPUs).

BlueField DPUs are programmed using DOCA [22, 23], which is currently in the limited early access phase; DPUs from other vendors have their own proprietary vendor-specific APIs. Despite the early access phase of DOCA, it is already being used for transparent malicious traffic detection and mitigation [24], routing [25], DNN training [26], and other applications are becoming possible with the evolution of its functionality.

Liu et. al. completed a performance study of BlueField-2 to understand how much processing power is available on the SmartNIC during transfers [27]. They found that, from a computational perspective, encryption, memory, and on-card processing operations on the SmartNIC perform significantly better than the general-purpose servers that were used for comparison. Additionally, they concluded that applications with a focus on these operations may be good candidates for the offloading to the SmartNIC.

While scientific software is seldom using encryption, various memory and on-card processing operations are performed. Therefore, in this paper we consider the potential for offload of molecular dynamics operations to (DPUs such as) NVIDIA BlueField SmartNICs.

The paper is organized as follows. First we describe the specialized computer systems for MD simulation, including

its specialized networking hardware. Afterwards we propose a possible approach to using DPUs for the acceleration of molecular dynamics simulation inspired by such specialized hardware. Finally, we conclude.

II. Specialized Hardware for Molecular Dynamics

Aside from the Anton supercomputer on which we will focus in this description of specialized hardware, other supercomputers specifically designed for MD simulation also exist. For example, MDGRAPE-4A, an improved version of MDGRAPE-4 [5], is a 512-node special-purpose machine built with one custom chip per node. Its capacity is not known, but it is possible to estimate its performance to be order of magnitude or two slower than Anton [4]. There is also FPGA-based specialized hardware [28, 29] with the performance comparable to single GPUs.

One of the most challenging problems remaining in supercomputing is, of course, the magnitude of the time frame within which certain MD simulations are performed. One always wants to find a compromise between the scale and speed needed to solve a research question and the practicality of the needed time frame to solve it.

While many users successfully used and still use general-purpose hardware for performing molecular dynamics simulations, there is also an effort from David E. Shaw Research to develop specialized hardware for molecular dynamics [3]. Two generations of Anton supercomputers, Anton 1 [30, 31, 32] and Anton 2 [33], were previously designed, built and implemented. When comparing them to the fastest general-purpose supercomputer of the time, the fact is that all of them reached multiple orders of magnitude more in terms of nanoseconds per day of MD simulations.

These increases in performance have translated into significant scientific progress. The development of Anton 3, a supercomputer designed around a new custom chip, once again increased the speed and scale of the required MD simulations for basic science and drug design [4]. In the following we focus on Anton 2 for which more information is available.

The goal is to perform each simulation step faster and therefore be able to reach longer timescales. To do so, Anton 2, a special-purpose parallel computer for molecular dynamics simulations, uses a combination of packet filtering in networking ASICs, intra-network reduction, and log-weight synchronization to reduce the communication requirements of MD simulations [2].

The quadratic computational complexity of the long-range Coulomb and van der Waals interaction makes its use impractical for larger systems. Therefore, it is important to maintain accuracy while ensuring that computational costs increase linearly with size. The common approach is to shorten interactions at some cutoff distance, Figure 1. This reduces the number of operations to $O(n)$ or $O(NlogN)$, particularly for the Coulomb interaction. Commonly used values in modern MD simulation software packages are in the range of $1-1.5$ nanometers.

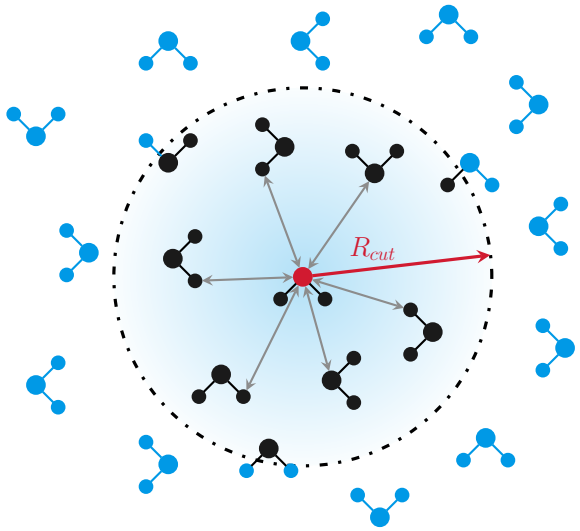

Fig. 1. (from [2]) The ball-and-stick icons represent water molecules. On every time step, pairwise (electrostatic and van der Waals) forces must be computed for all pairs of atoms up to a cutoff radius (R_{cut}). This requires communication when the simulation is distributed. The cutoff radius is illustrated for a single atom, with some of its interactions indicated explicitly. Note that all the interaction forces acting on the atom in question, are zero outside the cutoff radius.

In general, the chemical system is contained in a box with periodic boundary conditions which are unphysical. In that case, the system being simulated corresponds to infinite space, paved in all three dimensions with identical copies of the box. At each time step, the forces acting on each atom are calculated, where the positions and velocities of the atoms progress according to the classical laws of motion.

The simplest way to send an atom's position to its export set is with a sequence of unicasts, but this wastes both on- and off-chip bandwidth by communicating the position multiple times across individual network links (Fig. 2a).

Anton 2 improves the efficiency of this communication with a hardware mechanism for multicasting a position to an arbitrary set of nodes along with a specified set of channels (Fig. 2b). Following the computation, the sums of partial forces on atoms are returned to the source nodes. Generally, force reduction is done by taking a set of forces, specifically their numerical values, and conjuring up a smaller set by taking their sum in a specific way. In general-purpose supercomputers, this is done through the partial sum of the atom's node. Anton 2 uses in-network reductions to sum partial forces within the network, decreasing the cost of force return (Fig. 2d).

By summing the partial forces it receives and the locally computed forces, the atom's home node can obtain the total force acting on a specific atom.

On Anton 2, the approximation is to divide each home box into a set of sub boxes of identical size and construct export sets for the entire sub boxes. This allows export sets to be precomputed and converted to multicast patterns before the simulation begins.

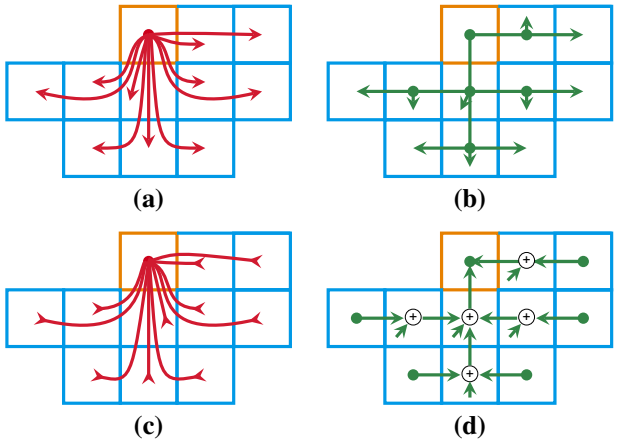

Fig. 2. (from [2]) a) When an atom's position is unicast to each node in the atom's export set, it must traverse several network links multiple times. b) On Anton 2, a hardware multicast mechanism eliminates this excess communication so that the position traverses each network link at most once. c) Partial forces for the atom are returned in the reverse direction. d) Anton 2 uses in-network reductions to sum partial forces within the network, again decreasing the amount of communication.

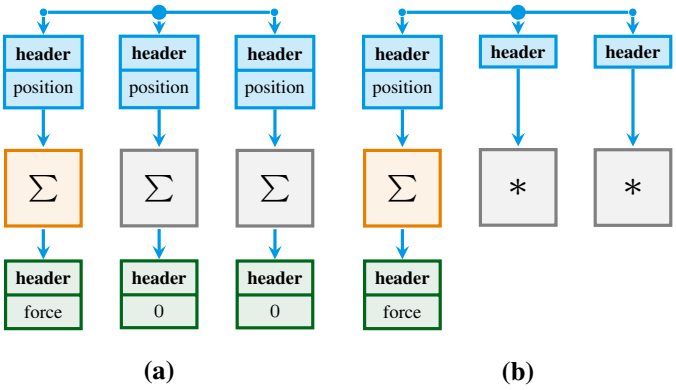

Fig. 3. (from [2]) a) A position packet is multicast to three nodes, only one of which is in the atom's precise export set. All three nodes compute and return partial forces, which will be zero for the two "extra" nodes. b) The position packet is filtered en route to the extra nodes. Only the header is sent, no computation is performed, and no forces are returned.

When the position of the atom is updated, the new position may fall into a new box, which requires communication across the network to update the simulation state at all the nodes. Therefore, a larger effective cutoff radius must be used to reduce atomic migration, further increasing the size of the import region. Also, the use of fine-grained sub boxes mitigates this effect by providing a better approximation to the exact import regions, however increasing the total number of sub boxes, resulting in greater bookkeeping and synchronization overheads.

The import synchronization mechanism requires that the position packet be delivered to all nodes within its export set, including additional nodes introduced by sub boxes and migration. Anton 2 implements position filtering by extending the multicast mechanism.

Most additional nodes do not actually need a position for any calculation. For example, the receiving node will try to calculate the interactions with the atom, only to find that all interactions are zero because the atom always lies outside the cutoff radius. The node must then return zero force to the atom, increasing the amount of force communication (Fig. 3a).

To solve this problem Anton 2 provides a position filtering mechanism to dynamically remove unnecessary positions from the network. Position filtering is an extension of Anton 2's network multicast mechanism that further optimizes this communication by dynamically cutting a multicast tree when it can be determined that certain endpoints do not require a copy of the atom's position (Fig. 3b).

The initial node of the atom must determine when all packets of atomic force have been received. When the number of force packets is known in advance, it is sufficient to simply count the packets. However, position filtering and reductions within the network create a synchronization problem: we no longer know how many force packets to expect for a given atom.

The solution to this synchronization problem is to assign a fixed "weight" to each atom. The weight is distributed between the atomic position packets and then transferred to its force packets after the force calculation. Instead of simply counting the force packets that are returned, their weights are added. The force of an atom fully accumulates after its full weight has been received (Fig. 4).

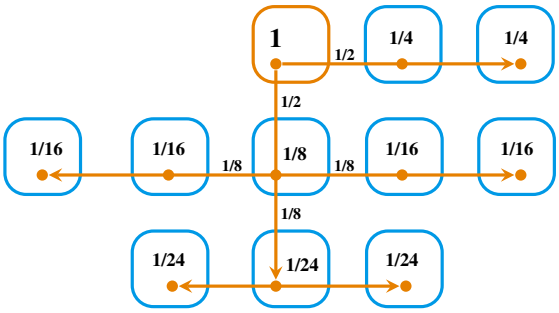

Fig. 4. (from [2]) Weight splitting. An atom's weight (1) is split at each multicast point. The weight delivered to a destination node varies depending on the node's location within the multicast tree. Weights are transferred to the returned force packets; all forces have been accumulated when their weights sum to 1.

Anton runs a specifically-developed proprietary molecular dynamics software Desmond [34].

Until the advent of DPUs, non-specialized supercomputers had no components comparable to Anton 2's network hardware. Now that DPUs are generally available, it is worth considering whether the approach used by Anton 2 can be applied on supercomputers with DPUs.

III. Molecular Dynamics Simulation Software Development for Data Processing Units

The present generation of DPUs, BlueField-2, provides 8 64-bit ARMv8 CPU cores with 1MB of L2 cache per 2

cores (4MB in total), 6MB of L3 cache, and 16GB or 32GB of onboard DDR4 RAM with error correction. The next-generation BlueField-3 DPU, originally announced in April 2021, is expected to sample the first quarter of 2022 [35]. This generational shift increases the on-chip processing power of the BlueField NIC from 8 CPU cores to a heterogeneous processor containing 16 CPU cores with larger cache than the previous generation and, in addition, a 16-core 256-thread accelerator for multi-threaded applications.

Applications for the BlueField DPUs are developed using DOCA [23], which provides Data Plane Development Kit (DPDK) for networking [19], similar to what CUDA provides for GPUs.

A. Overview of the DOCA Software Development Kit

The current version of DOCA software is 1.2 and is considered early access by NVIDIA and therefore made available only to select developers [22]. DOCA software, just like CUDA Toolkit, consists of an software development kit (SDK) and a runtime environment. The DOCA SDK offers several frameworks that are used to simplify the offload of computations to BlueField NICs using acceleration packages. The runtime environment is executing the application on the DPU. The SDK is divided into five distinct libraries [23], each with its own API:

- App Shield, which offers intrusion detection capabilities using the DPU services to collect data from the host's memory
- Deep packet inspection (DPI), which is a method of examining the full content of data packets as they traverse a monitored network checkpoint and provides a tool for enforcing network packet filtering
- Flow, which offers building generic execution pipes in hardware, where each pipe consists of packet match criteria and a set of actions that are taken on packets matching the criteria
- NetFlow, which simplifies the working with Cisco NetFlow packets
- Telemetry, which provides a way to collect user-defined data and send it to a telemetry service

While all of these libraries could be useful for data center applications, for the use of DOCA in scientific software we focus on the utilization of the Flow library.

B. DOCA Flow Library and Uses

Using the Flow library in DOCA, it is possible to develop hardware-accelerated packet processing. Packets can be modified or even dropped upon reception, as necessary. Packet headers (including TCP/UDP and IPv4/IPv6) can be processed in userspace, enabling the development of an MPI implementation on top of DOCA.

In particular, Bayatpour et. al. asked whether the features of SmartNICs can be utilized to reach pure communication performance while overlapping communication and computation for dense non-blocking collective communication, such as used by MPI [36]. They proposed BluesMPI which implements MPI Alltoall communication pattern [37] on DPUs: non-blocking collective operations are offloaded to a set of worker processes spawned to the ARM cores of BlueField, which communicate on the behalf of host processes.

C. Development of DPU Offload Library

To enable the offload of molecular dynamics to DPUs, we are proposing the development of a similar MPI library, potentially by extending an existing open-source implementation of MPI such as Open MPI, MPICH, or MVAPICH. We examined the usage of MPI in open-source MD simulation software packages GROMACS, HOOMD-blue, and LAMMPS to find out which MPI communication patterns are being utilized. In addition to Alltoall, to fulfill the requirements of the examined MD simulators, such a library will look into implementing the following MPI communication patterns: Send and Recv, Isend and Irecv, Wait, Test, Broadcast, Scatter, Gather, Reduce, and Scan. The library will be released under an open-source license so the other communication patterns can be added at a later date if needed.

This MPI library will be utilized by the molecular dynamics software for performing operations offloaded to the SmartNIC. Among the software described earlier, due to licensing and support for MPI, it is possible to implement this approach in GROMACS, HOOMD-blue, and LAMMPS. Due to its popularity, we propose using GROMACS for the development and testing of this approach. With the library being released under an open-source license, other software will be able to use it as well.

Initially, the target is the offload of the force reduction over the network, in which case DPUs perform a similar role as the networking ASICs in Anton 2. In addition, the benchmarks of BluesMPI [36] show that a parallel implementation of 3D fast Fourier transform (FFT), P3DFFT is sped up by 30% when DPUs are used. 3D FFT is an important part of particle mesh Ewald (PME), which is used for the computation of long-ranged components of the non-bonded interactions in molecular dynamics. Therefore, offloading this operation to the DPU is an interesting idea for the future.

IV. Conclusion

The authors of this paper are proposing the development of a novel DPU offload library implementing a subset of the MPI and using it to offload the force reduction in molecular dynamics to consumer hardware, as well as other operations in the future. The motivation for this development is providing molecular dynamics acceleration previously limited to Anton 2 supercomputer to a wide range of high-performance computers. The developments are planned in existing open-source projects (such as GROMACS and LAMMPS) so the implementation is widely available for future developments. The promised maintenance of backward compatibility between BlueField generations will, in particular, enable the proposed molecular dynamics computation offloading to benefit from the increased computing power.

ACKNOWLEDGMENT

This research was supported by grants from NVIDIA and utilized NVIDIA BlueField-2 data processing units (DPUs). This research was additionally supported by Horizon 2020 Framework Programme grant number 951732, European High-Performance Computing Joint Undertaking (EuroHPC JU) partnership's project National Competence Centres in the framework of EuroHPC (EuroCC).

The authors gratefully acknowledge the administrative and technical support from the Center for Advanced Computing and Modelling at the University of Rijeka. The authors are also thankful to Vedran Miletić and Dominik Kinkela of Group for Applications and Services on Exascale Research Infrastructure (GASERI) for fruitful discussions and support.

REFERENCES

[1] T. Hansson, C. Oostenbrink, and W. van Gunsteren, "Molecular dynamics simulations," *Current Opinion in Structural Biology*, vol. 12, no. 2, pp. 190–196, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959440X02003081

[2] J. Grossman, B. Towles, B. Greskamp, and D. E. Shaw, "Filtering, Reductions and Synchronization in the Anton 2 Network," in *2015 IEEE International Parallel and Distributed Processing Symposium*, May 2015, pp. 860–870, iSSN: 1530-2075.

[3] J. P. Grossman, J. S. Kuskin, J. A. Bank, M. Theobald, R. O. Dror, D. J. Ierardi, R. H. Larson, U. B. Schafer, B. Towles, C. Young, and D. E. Shaw, "Hardware support for fine-grained event-driven computation in Anton 2," *ACM SIGPLAN Notices*, vol. 48, no. 4, pp. 549–560, 2013. [Online]. Available: https://doi.org/10.1145/2499368.2451175

[4] D. E. Shaw, P. J. Adams, A. Azaria, J. A. Bank, B. Batson, A. Bell, M. Bergdorf, J. Bhatt, J. A. Butts, T. Correia, R. M. Dirks, R. O. Dror, M. P. Eastwood, B. Edwards, A. Even, P. Feldmann, M. Fenn, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, M. Gorlatova, B. Greskamp, J. Grossman, J. Gullingsrud, A. Harper, W. Hasenplaugh, M. Heily, B. C. Heshmat, J. Hunt, D. J. Ierardi, L. Iserovich, B. L. Jackson, N. P. Johnson, M. M. Kirk, J. L. Klepeis, J. S. Kuskin, K. M. Mackenzie, R. J. Mader, R. McGowen, A. McLaughlin, M. A. Moraes, M. H. Nasr, L. J. Nociolo, L. O'Donnell, A. Parker, J. L. Peticolas, G. Pocina, C. Predescu, T. Quan, J. K. Salmon, C. Schwink, K. S. Shim, N. Siddique, J. Spengler, T. Szalay, R. Tabladillo, R. Tartler, A. G. Taube, M. Theobald, B. Towles, W. Vick, S. C. Wang, M. Wazlowski, M. J. Weingarten, J. M. Williams, and K. A. Yuh, "Anton 3: twenty microseconds of molecular dynamics simulation before lunch," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. St. Louis Missouri: ACM, Nov. 2021, pp. 1–11. [Online]. Available: https://dl.acm.org/doi/10.1145/3458817.3487397

[5] I. Ohmura, G. Morimoto, Y. Ohno, A. Hasegawa, and M. Taiji, "Mdgrape-4: a special-purpose computer system for molecular dynamics simulations," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2021, p. 20130387, 2014.

[6] I. Buck, "Gpu computing: Programming a massively parallel processor," in *International Symposium on Code Generation and Optimization (CGO'07)*. IEEE, 2007, pp. 17–17.

[7] "GPU Applications | High Performance Computing." [Online]. Available: https://www.nvidia.com/en-us/gpu-accelerated-applications/

[8] W. Liu, B. Schmidt, G. Voss, and W. Müller-Wittig, "Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA," *Computer Physics Communications*, vol. 179, no. 9, pp. 634–641, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465508002191

[9] C. Kutzner, S. Páll, M. Fechner, A. Esztermann, B. L. de Groot, and H. Grubmüller, "Best bang for your buck: GPU nodes for GROMACS biomolecular simulations," *Journal of Computational Chemistry*, vol. 36, no. 26, pp. 1990–2008, 2015, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.24030. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.24030

[10] N. Jouppi, C. Young, N. Patil, and D. Patterson, "Motivation for and Evaluation of the First Tensor Processing Unit," *IEEE Micro*, vol. 38, no. 3, pp. 10–19, May 2018, conference Name: IEEE Micro.

[11] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman, "AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules," *Computer Physics Communications*, vol. 91, no. 1-3, pp. 1–41, 1995, publisher: Elsevier.

[12] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. a. Swaminathan, and M. Karplus, "CHARMM: a program for macromolecular energy, minimization, and dynamics calculations," *Journal of computational chemistry*, vol. 4, no. 2, pp. 187–217, 1983, publisher: Wiley Online Library.

[13] H. J. Berendsen, D. van der Spoel, and R. van Drunen, "GROMACS: A message-passing parallel molecular dynamics implementation," *Computer physics communications*, vol. 91, no. 1-3, pp. 43–56, 1995, publisher: Elsevier.

[14] J. A. Anderson, J. Glaser, and S. C. Glotzer, "HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations," *Computational Materials Science*, vol. 173, p. 109363, 2020, publisher: Elsevier.

[15] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in't Veld, A. Kohlmeyer, S. G. Moore, and T. D. Nguyen, "LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Computer Physics Communications*, vol. 271, p. 108171, 2022, publisher: Elsevier.

[16] P. Eastman and V. Pande, "OpenMM: A hardware-independent framework for molecular simulations," *Computing in science & engineering*, vol. 12, no. 4, pp. 34–39, 2010, publisher: IEEE.

[17] J. A. Rackers, Z. Wang, C. Lu, M. L. Laury, L. Lagardère, M. J. Schnieders, J.-P. Piquemal, P. Ren, and J. W. Ponder, "Tinker 8: Software Tools for Molecular Design," *Journal of Chemical Theory and Computation*, vol. 14, no. 10, pp. 5273–5289, Oct. 2018, publisher: American Chemical Society. [Online]. Available: https://doi.org/10.1021/acs.jctc.8b00529

[18] V. S. Pande, K. Beauchamp, and G. R. Bowman, "Everything you wanted to know about markov state models but were afraid to ask," *Methods*, vol. 52, no. 1, pp. 99–105, 2010.

[19] "Data Plane Development Kit." [Online]. Available: https://www.dpdk.org/

[20] "Storage Performance Development Kit." [Online]. Available: https://spdk.io/

[21] D. Gallatin, "Serving Netflix Video at 400Gb/s on FreeBSD :: FreeBSD Presentations and Papers," 2021. [Online]. Available: https://papers.freebsd.org/2021/eurobsdcon/gallatin-netflix-freebsd-400gbps/

[22] "NVIDIA DOCA Software Framework," Oct. 2020. [Online]. Available: https://developer.nvidia.com/networking/doca

[23] "NVIDIA DOCA SDK Documentation." [Online]. Available: https://docs.nvidia.com/doca/sdk/

[24] J. Ermerins, W. Bakker, and C. Both, "Transparent malicious traffic detection and mitigation using the Nvidia BlueField DPU," University of Amsterdam, Tech. Rep., 2021.

[25] "Developing Applications with NVIDIA BlueField DPU and NVIDIA DOCA Libraries," Feb. 2022. [Online]. Available: https://developer.nvidia.com/blog/developing-applications-with-bluefield-dpu-and-doca-libraries/

[26] A. Jain, N. Alnaasan, A. Shafi, H. Subramoni, and D. K. Panda, "Accelerating CPU-based Distributed DNN Training on Modern HPC Clusters using BlueField-2 DPUs," in *2021 IEEE Symposium on High-Performance Interconnects (HOTI)*, Aug. 2021, pp. 17–24, iSSN: 2332-5569.

[27] J. Liu, C. Maltzahn, C. Ulmer, and M. L. Curry, "Performance Characteristics of the BlueField-2 SmartNIC," *arXiv:2105.06619 [cs]*, May 2021, arXiv: 2105.06619. [Online]. Available: http://arxiv.org/abs/2105.06619

[28] M. Schaffner and L. Benini, "On the feasibility of fpga acceleration of molecular dynamics simulations," *arXiv preprint arXiv:1808.04201*, 2018.

[29] C. Yang, T. Geng, T. Wang, R. Patel, Q. Xiong, A. Sanaullah, C. Wu, J. Sheng, C. Lin, V. Sachdeva *et al.*, "Fully integrated fpga molecular dynamics simulations," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–31.

[30] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao *et al.*, "Anton, a special-purpose machine for molecular dynamics simulation," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 1–12, 2007.

[31] ——, "Anton, a special-purpose machine for molecular dynamics simulation," *Communications of the ACM*, vol. 51, no. 7, pp. 91–97, 2008.

[32] D. E. Shaw, R. O. Dror, J. K. Salmon, J. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers *et al.*, "Millisecond-scale molecular dynamics simulations on anton," in *Proceedings of the conference on high performance computing networking, storage and analysis*, 2009, pp. 1–11.

[33] D. E. Shaw, J. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton *et al.*, "Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer," in *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2014, pp. 41–53.

[34] K. J. Bowers, D. E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, and F. D. Sacerdoti, "Scalable algorithms for molecular dynamics simulations on commodity clusters," in *SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*. IEEE, 2006, pp. 43–43.

[35] "NVIDIA Extends Data Center Infrastructure Processing Roadmap with BlueField-3." [Online]. Available: http://nvidianews.nvidia.com/news/nvidia-extends-data-center-infrastructure-processing-roadmap-with-bluefield-3

[36] M. Bayatpour, N. Sarkauskas, H. Subramoni, J. Maqbool Hashmi, and D. K. Panda, "BluesMPI: Efficient MPI Non-blocking Alltoall Offloading Designs on Modern BlueField Smart NICs," in *High Performance Computing*, ser. Lecture Notes in Computer Science, B. L. Chamberlain, A.-L. Varbanescu, H. Ltaief, and P. Luszczek, Eds. Cham: Springer International Publishing, 2021, pp. 18–37.

[37] L. Clarke, I. Glendinning, and R. Hempel, "The MPI message passing interface standard," in *Programming environments for massively parallel distributed systems*. Springer, 1994, pp. 213–218.