

Early evaluation of the hybrid cluster with torus interconnect aimed at cost-effective molecular-dynamics simulations

Vladimir V. Stegailov¹, Alexander Agarkov², Sergey Biryukov²,
Timur Ismagilov², Nikolay Kondratyuk^{1,3,4}, Evgeny Kushtanov²,
Dmitry Makagon², Anatoly Mukosey², Alexander Semenov², Alexey Simonov²,
Vyacheslav Vecher^{1,3}

¹ Joint Institute for High Temperatures of RAS, Moscow, Russia

² NICEVT, Moscow, Russia

³ Moscow Institute of Physics and Technology, Dolgoprudny, Russia

⁴ National Research University Higher School of Economics, Moscow, Russia

v.stegailov@hse.ru

Abstract. In this paper, we describe the Desmos cluster that consists of 32 hybrid nodes connected by a low-latency high-bandwidth torus interconnect. This cluster is aimed at cost-effective classical molecular dynamics calculations. We present strong scaling benchmarks for GROMACS and LAMMPS and compare the results with other HPC systems. This cluster serves as a test bed for the Angara interconnect and verifies its ability to unite MPP systems speeding-up effectively MPI-based applications. The interconnect is based on the Angara NIC that supports 3D and 4D torus network topologies. We describe the interconnect presenting typical MPI benchmarks.

1 Introduction

Rapid development of parallel computational methods and supercomputer hardware provide great benefits for atomistic simulation methods. At the moment, these mathematical models and computational codes are not only the tools of fundamental research but the more and more intensively used instruments for diverse applied problems [1]. For classical molecular dynamics (MD) the limits of the system size and the simulated time are trillions of atoms [2] and milliseconds [3] (i.e. 10^9 steps with a typical MD step of 1 femtosecond).

There are two mainstream ways of MD acceleration. The first one is the use of distributed memory massively-parallel programming (MPP) systems. For MD calculations, domain decomposition is a natural technique to distribute both the computational load and the data across nodes of MPP systems [4]. Corresponding load-balancing problems can be usually solved very efficiently (e.g. [5]).

The second possibility consists in the increase of the computing capabilities of individual nodes of MPP systems. Multi-CPU and multi-core shared-memory node architectures provide essential acceleration. However, the scalability of shared memory systems is limited by their cost and speed limitations of

DRAM access for multi-socket and/or multi-core nodes. It is the development of GPGPU technologies that boosts significantly the performance of shared-memory systems.

This year is the 10th anniversary of Nvidia CUDA technology that was introduced in 2007 and provided a convenient technique for GPU programming. Many computational algorithms have been rewritten and thoroughly optimized to use the GPU capabilities. However, the majority of them deploy only a fraction of the GPU theoretical performance even after careful tuning, e.g. see [6,7,8]. The sustained performance is usually limited by the memory-bound nature of the algorithms.

Among GPU-aware MD software one can point out GROMACS [9] as, perhaps, the most computationally efficient MD tool and LAMMPS [10] as one of the most versatile and flexible for MD models creation. Different GPU off-loading schemes were implemented in LAMMPS [11,12,13,14]. GROMACS provides a highly optimized GPU-scheme as well [15].

There are other ways to increase performance of individual nodes: using GPU accelerators with OpenCL, using Intel Xeon Phi accelerators or even using custom built chips like MDGRAPE [16] or ANTON [3]. Currently, however, general purpose Nvidia GPUs provide the most cost-effective way for high performance MD calculations [17].

Modern MPP systems can unite up to 10^5 nodes for solving one computational problem. For this purpose, MPI is the most widely used programming model. The architecture of the individual nodes can differ significantly and is usually selected (co-designed) for the main type of MPP system deployment. The most important component of MPP systems is the interconnect that properties stand behind the scalability of any MPI-based parallel algorithm.

In this work, we describe the Desmos computing cluster that is based on cheap 1CPU+1GPU nodes connected by an original Angara interconnect with torus topology. We describe this novel interconnect and the resulting performance of the cluster for MD models implemented in GROMACS and LAMMPS.

2 Related work

Torus topologies of the interconnect has several attractive aspects in comparison with fat-tree topologies. In 1990s, the development of MPP systems has its peak during the remarkable success of Cray T3E systems based on the 3D torus interconnect topology [18]. Cray T3E was the first supercomputer that provided 1 TFlops of sustained performance. In June 1998 Cray T3E occupied 4 of top-5 records of the Top500 list. In 2004, after several years of the dominance of Beowulf clusters, a custom-built torus interconnect appeared in the IBM BlueGene/L supercomputer [19]. Subsequent supercomputers of Cray and IBM had torus interconnects as well (with the exception of the latest Cray XC series). Fujitsu designed K Computer based on the Tofu torus interconnect [20]. The AURORA Booster and GREEN ICE Booster supercomputers are based on the EXTOLL torus interconnect [21].

Among the references to the recent developments of original types of supercomputer interconnects in Russia, we can mention the MVS-Express interconnect based on the PCI-Express bus [22], the FPGA prototypes of the SKIF-Aurora [23] and Pautina [24] torus interconnects. Up to this moment the Angara interconnect has been evolving all the way from the FPGA prototype [25,26] to the ASIC-based card [27].

MD benchmarks are among the most popular tests for supercomputers. For example, ApoA1 (protein in water) is a widespread model and we can find the corresponding benchmark data for Cray XK6 [28], for IBM BlueGene/P and BlueGene/Q systems [29], for K Computer [30] (in all cases calculations were made with NAMD). The AURORA Booster supercomputer was benchmarked using LAMMPS [21].

There are examples of the development of alternative highly optimized MD codes for supercomputers. The MD code ls1 mardyn was optimized for Intel x86 CPUs including vectorization and shared-memory parallelization that allowed to simulate the multi-trillion atoms Lennard-Jones liquid model on 146016 cores of the SuperMUC supercomputer [2].

Torus topology is believed to be beneficial for strong scaling of many parallel algorithms. However, the accurate data that verify this assumption are quite rare. Probably, the most extensive work was done by Fabiano Corsetti who compared torus and fat tree topologies using the SIESTA electronic structure code and six large-scale supercomputers belonging to the PRACE Tier-0 network [31]. The author concluded that machines implementing torus topologies demonstrated a better scalability to large system sizes than those implementing fat tree topologies. The comparison of the benchmark data for CP2k showed a similar trend [7].

3 The Desmos Cluster

The hardware for the cluster was selected in order to maximize (within the budget limit) the number of nodes and the efficiency of a single node for MD workloads. The resulting node configuration is shown in Table 1.

The Nvidia GeForce 1070 cards have no error-correcting code (ECC) memory in contrast to professional accelerators. For this reason, it was necessary to make

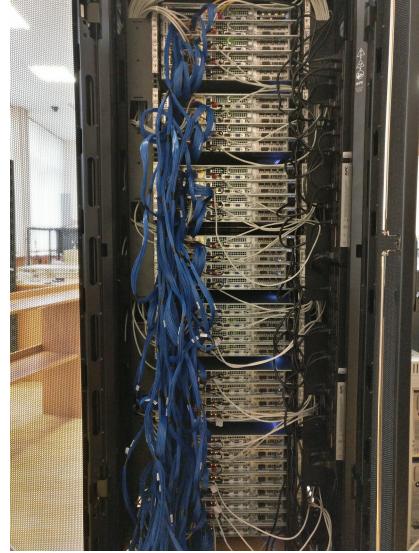


Fig. 1: The Desmos cluster.

Table 1: Components of a single node (Moscow prices for October 2016)

Chassis	Supermicro SuperServer 1018GR-T	955 USD
CPU	Intel Xeon E5-1650v3 (6 cores, 3.5 GHz)	700 USD
GPU	Nvidia GeForce GTX 1070 (8 GB GDDR5)	550 USD
DRAM	DDR4 8 GB	70 USD
Storage	HDD 500 GB	44 USD
	Total price	2319 USD

sure that there is no hardware memory errors in each GPU. Testing of each GPU was performed using MemtestG80 [32] during more than 4 hours for each card. No errors were detected for 32 cards considered.

The cooling of the GPU cards was a special question. We used ASUS GeForce GTX 1070 8GB Turbo Edition. Each card was partially disassembled prior to installation into 1U chassis. The plastic cover and the dual-ball bearing fan were removed that made the card suitable for horizontal air flow cooling inside chassis.

The nodes are connected by Gigabit Ethernet and Angara interconnect (in the 4D-torus $4 \times 2 \times 2 \times 2$). Due to budget limitations we did not use all possible ports for the full 4D-torus topology. The currently implemented topology is 4D-torus $4 \times 2 \times 2 \times 2$ ($X \times Y \times Z \times K$) but each node along Y, Z, K dimensions is connected to another node by one link only.

There is a front-end node with the same configuration as all 32 computing nodes of the cluster (the front-end node is connected to GE only). The cluster is running under SLES 11 SP4 with Angara MPI (based on MPICH 3.0.4).

The cluster energy consumption is 6.5 kW in the idle state and 14.4 kW under full load.

4 The Angara Interconnect

The Angara interconnect is a Russian-designed communication network with torus topology. The interconnect ASIC was developed by JSC NICEVT and manufactured by TSMC with the 65 nm process.

The Angara architecture uses some principles of IBM Blue Gene L/P and Cray Seastar2/Seastar2+ torus interconnects. The torus interconnect developed by EXTOLL is a similar project [21]. The Angara chip supports deadlock-free adaptive routing based on bubble flow control [33], direction ordered routing [18,19] and initial and final hops for fault tolerance [18].

Each node has a dedicated memory region available for remote access from other nodes (read, write, atomic operations) to support the OpenSHMEM and PGAS languages. Multiple parallel programming models are supported MPI and OpenMP including.

Table 2: Comparison of the Desmos cluster with the Polytekhnik cluster used as a reference for MPI benchmarks

Cluster	Desmos	Polytekhnik
Chassis	SuperServer 1018GR-T	RSC Tornado
Processor	E5-1650v3 (6c, 3.0 GHz)	2 x E5-2697v3 (14c, 2.6 GHz)
GPU	Nvidia GeForce GTX 1070	—
Memory	DDR4 8 GB	DDR4 64 GB
Number of nodes	32	612
Interconnect	Angara 4D-torus $4 \times 2 \times 2 \times 2$	Infiniband 4x FDR 2:1
Operating system	SLES 11 SP4	CentOS 7.0.1406
Compiler	Intel Parallel Studio XE 2017	Intel Parallel Studio XE 2016
MPI	Angara MPI (based on MPICH 3.0.4)	Intel MPI 5.1.2

The network adapter is a PCI Express extension card that is connected to the adjacent nodes by up to 6 cables (or up to 8 with an extension card). The following topologies are supported: a ring, 2D, 3D and 4D tori.

To provide more insights into Angara communication behavior we present a performance evaluation comparison of the Desmos cluster and the Polytekhnik cluster with the Mellanox Infiniband 4x FDR 2:1 blocking interconnect. Table 2 compares two systems. Both systems are equipped with Haswell CPUs but the processors characteristics differ very much.

Figure 2a shows the latency results obtained by the OSU Micro-Benchmarks test. Angara has extremely low latency $0.85 \mu\text{s}$ for the 16 bytes message size and exceeds 4x FDR interconnect for all small message sizes.

We use Intel MPI Benchmarks to evaluate MPI_Barrier and MPI_Alltoall operation times for different number of nodes of the Desmos and Polytekhnik clusters. The Angara superiority for small messages explains better results for MPI_Barrier (Figure 2b) and MPI_Alltoall with 16 byte messages (Figure 2c). For large messages (256 Kbytes) the Desmos results are worse than that of the Polytechnik cluster (Figure 2d). It can be explained by loose connectivity of the Desmos torus topology (2.5 links per node) and the performance weakness of the current variant of the Angara MPI implementation.

5 MD benchmarks

Nowadays, there is no novelty in the partial use of single precision in MD calculations with consumer-grade GPUs. The results of such projects as Folding@Home

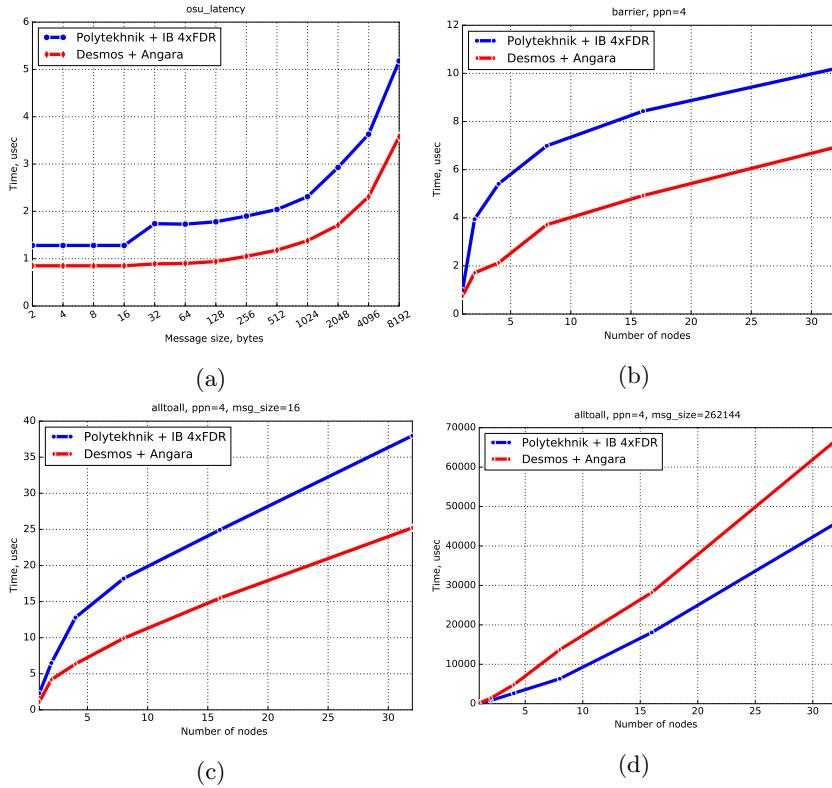


Fig. 2: The OSU latency between two adjacent nodes (a). Times for MPI_BARRIER with 4 processes per node (b). Times for MPI_Alltoall with 4 processes per node and message sizes 16 b (c) and 256 Kb (d).

confirmed the broad applicability of this approach. Recent developments of optimized MD algorithms include the validation of the single precision solver (e.g. [34]). In this study we do not consider the questions of accuracy and limit ourselves to the benchmarks of computational efficiency.

The first set of benchmark data is shown on Figure 3. It illustrates the efficiency of LAMMPS, LAMMPS with the GPU package (with mixed precision) or LAMMPS with USER-INTEL package running on one computational node for different numbers of atoms in the MD model. The USER-INTEL package provides SIMD-optimized versions of certain interatomic potentials in LAMMPS. As expected, the CPU+GPU pair shows maximum performance for sufficiently large system sizes. The results for the Desmos node are compared with the results for a two-socket node with 14-core Haswell CPUs (the MVS1P5 cluster).

We see that for the simple Lennard-Jones (LJ) liquid benchmark the GPU-version of LAMMPS on one Desmos node provides two times higher times-to-solution than the SIMD-optimized LAMMPS on 2 x 14-core Haswell CPUs.

However for the liquid $C_{30}H_{62}$ oil benchmark the GPU-version of LAMMPS is faster starting from 100 thousand atoms per node.

For comparison, we present the results for the same benchmark on the professional grade Nvidia Tesla K80 card.

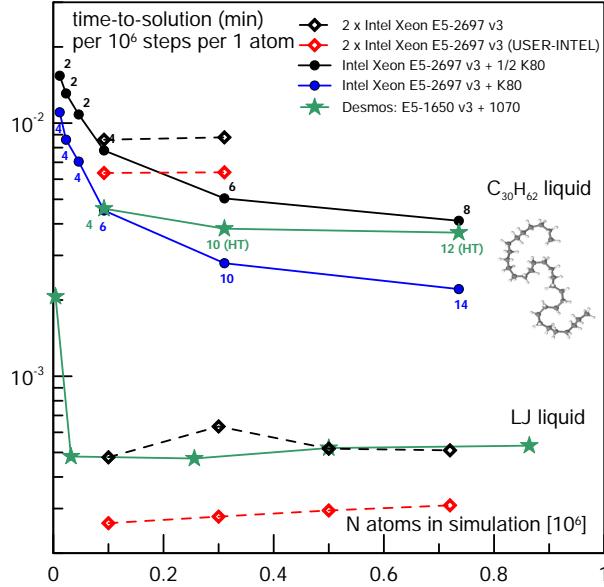


Fig. 3: Times-to-solution per atom per timestep for Lennard-Jones liquid and $C_{30}H_{62}$ oil LAMMPS MD models of different size and different hardware combinations.

An apolipoprotein A1 (ApoA1) in water MD model is a popular benchmark system with $\sim 100k$ atoms. Figure 4 shows the weak scaling results for the LJ system, for the $C_{30}H_{62}$ oil model and for the ApoA1 system. We see that weak scaling becomes better for larger models.

We should mention that weak scaling could be slightly improved after implementation of the topology-aware cartesian MPI-communicators in Angara MPI.

The “time-to-solution” criterion leads us to the evident choice of a time for one MD integration step as one parameter for the metric. The second parameter should characterize the hardware. Usually the number of some abstract processing elements (e. g. cores) is considered. However although this metric serves well in the *weak* and *strong* scaling benchmarks for the given system, it does not allow to compare essentially different hardware. In order to overcome this problem we consider the total peak performance R_{peak} as a second parameter for the metric that put on equal footing all HPC hardware under consideration [7].

Since the Desmos cluster rely heavily on single-precision performance we need to devise some reasonable conversion of SP Flops to DP Flops. Here we take $R_{peak}^{node} = 6 * 3.5 * 16 + 0.5 * R_{peak}^{GPU-SP} = 336 + 2892 = 3228$ GFlops.

The performance of different clusters based on ApoA1 test is shown in Figure 5 in terms of seconds per 1 atom for 1 MD step and the declared peak performance. The dotted line shows ideal scalability with performance 0.1 MFlops/atom/step.

Despite different architectures and MD software, these data can be used to analyze the efficiency of supercomputer systems for biomolecular MD studies. The black and green lines with dots show the performance of the CPU cluster with Intel Xeon nodes [6]. The new Desmos cluster (the purple filled stars) with GROMACS shows better scalability than Cray XK6 [28] and the K Computer with NAMD. It shows better efficiency than the benchmarks for BlueGene systems[29].

The green stars show the 27 times replicated ApoA1 system benchmark for the Desmos cluster. These data show the continuous scaling with larger number atoms per node. It indicates the possibility for adding more nodes to Desmos without noticeable efficiency degradation.

The work [17] gives very instructive guidelines for achieving the best performance for the minimal price in 2015. Authors compared different configurations of clusters using two biological benchmarks: the membrane channel protein embedded in a lipid bilayer surrounded by water (MEM, ~100k atoms) and the bacterial ribosome in water with ions (RIB, ~2M atoms). The GROMACS package was used for all tests. We follow the same protocol but use longer MD simulations than in [17] to obtain the best performance.

We compare the results obtained on Desmos cluster with the best choice of [17]: the nodes that consist of 2 socket Xeon E5-2670 v2 with 2 780Ti and connected via InfiniBand QDR. The costs of hardware in Euros are displayed on Y axis (for price conversion we use the ratio Euro/USD=1.1 for October 2016) and the performance in ns/day is shown on X axis in Figure 6. The numbers show the number of nodes used. The grey lines indicate ideal scaling.

Desmos (red color) shows better strong scaling for the MEM benchmark than the best system configuration provided by Kutzner et al. (blue color) [17]. The major reason is, of course, the new Pascal GPU architecture. The saturation is achieved after 20 nodes which corresponds to the small amount of atoms per node, below the GPU efficiency threshold. In the case of the RIB benchmark, Desmos demonstrates ideal scaling after 16 nodes which shows that the productivity can be increased with larger number of nodes.

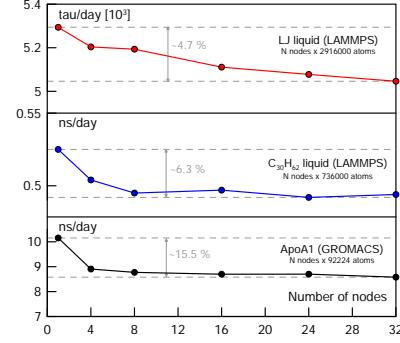


Fig. 4: Weak scaling benchmarks for the Desmos cluster and three different MD models.

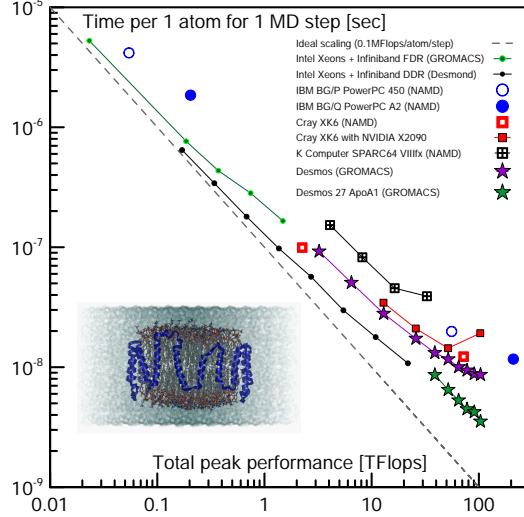


Fig. 5: ApoA1 benchmark results obtained with GROMACS for the Desmos cluster (including the 3x3x3 replicated model). For comparison we show the data for the same benchmark but obtained with NAMD for different supercomputers: Cray XK6 [28], IBM BlueGene/P and BlueGene/Q [29], K Computer [30].

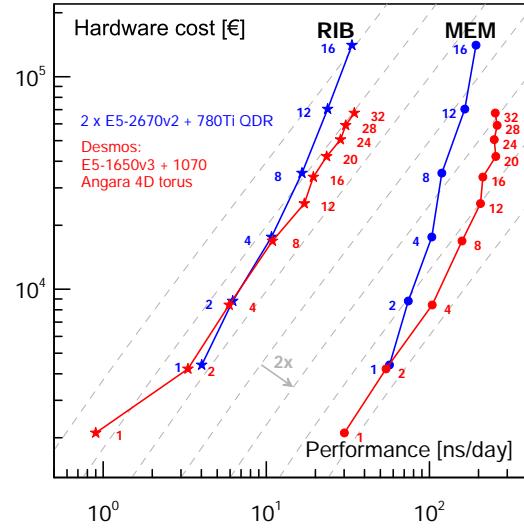


Fig. 6: The cost of hardware vs achieved performance for MEM and RIB benchmarks. The dashed lines show ideal scaling. Desmos results are compared with the published results [17].

6 Conclusions

In the paper we described the cost-effective Desmos cluster targeted to MD calculations. The results of this work confirmed the high efficiency of commodity GPU hardware for MD simulations. The Desmos cluster is the first application of the Angara interconnect for a GPU-based MPP system. The features of the Angara interconnect provided very reasonable level of efficiency for the MPP system considered. The MPI benchmarks presented supported the competitive level of this network for HPC applications.

The results of the work were obtained using computational resources of Peter the Great Saint-Petersburg Polytechnic University Supercomputing Center (<http://www.spbstu.ru>). The authors are grateful to the Forsite company for the access to the server with Nvidia Tesla K80. The authors acknowledge Joint Supercomputer Centre of Russian Academy of Sciences (<http://www.jscc.ru>) for the access to the MVS-10P supercomputer.

The work of the JIHT team (V.S., N.K. and V.V.) was supported by the grant No. 14-50-00124 of the Russian Science Foundation. Their work included the development of the Desmos cluster architecture, tuning of the MD codes performance and MD benchmarks. The NICEVT team developed the Angara interconnect and the corresponding low-level software stack, built and tuned the Desmos cluster.

Comment for reviewers. Unfortunately, the Angara interconnect bandwidth is the subject of the non-disclosure agreement of the Angara project.

References

1. Alexander Heinecke, Wolfgang Eckhardt, Martin Horsch, and Hans-Joachim Bungartz. *Supercomputing for Molecular Dynamics Simulations*. Springer, 2015.
2. Wolfgang Eckhardt, Alexander Heinecke, Reinhold Bader, Matthias Brehm, Nicolay Hammer, Herbert Huber, Hans-Georg Kleinhenz, Jadran Vrabec, Hans Hasse, Martin Horsch, Martin Bernreuther, Colin W. Glass, Christoph Niethammer, Arndt Bode, and Hans-Joachim Bungartz. 591 TFLOPS multi-trillion particles simulation on SuperMUC. In Julian Martin Kunkel, Thomas Ludwig, and Hans Werner Meuer, editors, *Supercomputing*, volume 7905 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2013.
3. Stefano Piana, John L Klepeis, and David E Shaw. Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Current Opinion in Structural Biology*, 24(0):98 – 105, 2014.
4. Godehard Sutmann. Classical molecular dynamics. In J. Grotendorst, D. Marx, and A. Muramatsu, editors, *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*, volume 10 of *NIC Series*, pages 211–254. John von Neumann Institute for Computing, Julich, 2002.
5. Christoph Begau and Godehard Sutmann. Adaptive dynamic load-balancing with irregular domain decomposition for particle simulations. *Computer Physics Communications*, 190:51 – 61, 2015.

6. G. S. Smirnov and V. V. Stegailov. Efficiency of classical molecular dynamics algorithms on supercomputers. *Mathematical Models and Computer Simulations*, 8(6):734–743, 2016.
7. Vladimir V. Stegailov, Nikita D. Orekhov, and Grigory S. Smirnov. *Parallel Computing Technologies: 13th International Conference, PaCT 2015, Petrozavodsk, Russia, August 31-September 4, 2015, Proceedings*, chapter HPC Hardware Efficiency for Quantum and Classical Molecular Dynamics, pages 469–473. Springer International Publishing, Cham, 2015.
8. Krzysztof Rojek, Roman Wyrzykowski, and Lukasz Kuczynski. Systematic adaptation of stencil-based 3D MPDATA to GPU architectures. *Concurrency and Computation: Practice and Experience*, pages n/a–n/a, 2016.
9. H.J.C. Berendsen, D. van der Spoel, and R. van Drunen. Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(13):43 – 56, 1995.
10. Steve Plimpton. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.*, 117(1):1–19, March 1995.
11. C. R. Trott, L. Winterfeld, and P. S. Crozier. General-purpose molecular dynamics simulations on GPU-based clusters. *ArXiv e-prints*, September 2010.
12. W. Michael Brown, Peng Wang, Steven J. Plimpton, and Arnold N. Tharrington. Implementing molecular dynamics on hybrid high performance computers — short range forces. *Computer Physics Communications*, 182(4):898 – 911, 2011.
13. W. Michael Brown, Axel Kohlmeyer, Steven J. Plimpton, and Arnold N. Tharrington. Implementing molecular dynamics on hybrid high performance computers — Particle-particle particle-mesh. *Computer Physics Communications*, 183(3):449 – 459, 2012.
14. H. Carter Edwards, Christian R. Trott, and Daniel Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, 74(12):3202 – 3216, 2014. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing.
15. Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 12:19 – 25, 2015.
16. Itta Ohmura, Gentaro Morimoto, Yousuke Ohno, Aki Hasegawa, and Makoto Taiji. MDGRAPE-4: a special-purpose computer system for molecular dynamics simulations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2021), 2014.
17. Carsten Kutzner, Szilard Pall, Martin Fechner, Ansgar Esztermann, Bert L. de Groot, and Helmut Grubmüller. Best bang for your buck: Gpu nodes for gromacs biomolecular simulations. *Journal of Computational Chemistry*, 36(26):1990–2008, 2015.
18. Steven L. Scott and Gregory M. Thorson. The Cray T3E network: Adaptive routing in a high performance 3D torus. In *HOT Interconnects IV, Stanford University, August 15-16, 1996*, 1996.
19. N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas. Blue Gene/L torus interconnection network. *IBM J. Res. Dev.*, 49(2):265–276, March 2005.
20. Y. Ajima, T. Inoue, S. Hiramoto, Y. Takagi, and T. Shimizu. The Tofu interconnect. *IEEE Micro*, 32(1):21–31, Jan 2012.

21. S. Neuwirth, D. Frey, M. Nuessle, and U. Bruening. Scalable communication architecture for network-attached accelerators. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 627–638, Feb 2015.
22. G. S. Elizarov, V. S. Gorbunov, V. K. Levin, A. O. Latsis, V. V. Korneev, A. A. Sokolov, D. V. Andryushin, and Yu. A. Klimov. Communication fabric MVS-Express. *Vychisl. Metody Programm.*, 13(3):103–109, 2012.
23. I. A. Adamovich, A. V. Klimov, Yu. A. Klimov, A. Yu. Orlov, and A. B. Shvorin. Thoughts on the development of SKIF-Aurora supercomputer interconnect. *Programmnye Sistemy: Teoriya i Prilozheniya*, 1(3):107–123, 2010.
24. Yu. A. Klimov, A. B. Shvorin, A. Yu. Khrenov, I. A. Adamovich, A. Yu. Orlov, S. M. Abramov, Yu. V. Shevchuk, and A. Yu. Ponomarev. Pautina: the high performance interconnect. *Programmnye Sistemy: Teoriya i Prilozheniya*, 6(1):109–120, 2015.
25. A. A. Korzh, D. V. Makagon, A. A. Borodin, I. A. Zhabin, E. R. Kushtanov, E. L. Syromyatnikov, and E. V. Cheryomushkina. Russian 3D-torus interconnect with globally addressable memory support. *Vestnik YuUrGU. Ser. Mat. Model. Progr.*, (6):41–53, 2010.
26. A. V. Mukosey, A. S. Semenov, and A. S. Simonov. Simulation of collective operations hardware support for Angara interconnect. *Vestn. YuUrGU. Ser. Vych. Matem. Inform.*, 4(3):40–55, 2015.
27. A. A. Agarkov, T. F. Ismagilov, D. V. Makagon, A. S. Semenov, and A. S. Simonov. Performance evaluation of the Angara interconnect. In *Proceedings of the international conference “Russian Supercomputing Days” — 2016*, pages 626–639, 2016.
28. Yanhua Sun, Gengbin Zheng, Chao Mei, Eric J. Bohm, James C. Phillips, Laximant V. Kalé, and Terry R. Jones. Optimizing fine-grained communication in a biomolecular simulation application on Cray XK6. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC ’12, pages 55:1–55:11, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.
29. S. Kumar, Y. Sun, and L. V. Kalé. Acceleration of an asynchronous message driven programming paradigm on IBM Blue Gene/Q. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, pages 689–699, May 2013.
30. Akinori Yonezawa. “Concurrent objects, agents and HPC,” AICS RIKEN, 2012. URL: <http://www.slideshare.net/AkinoriYonezawa/ageretalk>.
31. Fabiano Corsetti. Performance analysis of electronic structure codes on HPC systems: A case study of SIESTA. *PLOS ONE*, 9(4):1–8, 04 2014.
32. Imran S. Haque and Vijay S. Pande. Hard data on soft errors: A large-scale assessment of real-world error rates in GPGPU. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID ’10*, pages 691–696, Washington, DC, USA, 2010. IEEE Computer Society.
33. V. Puente, R. Beivide, J. A. Gregorio, J. M. Prellezo, J. Duato, and C. Izu. Adaptive bubble router: a design to improve performance in torus networks. In *Proceedings of the 1999 International Conference on Parallel Processing*, pages 58–67, 1999.
34. Markus Höhnerbach, Ahmed E. Ismail, and Paolo Bientinesi. The vectorization of the Tersoff multi-body potential: An exercise in performance portability. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC ’16, pages 7:1–7:13, Piscataway, NJ, USA, 2016. IEEE Press.