






Early Performance Evaluation of the Hybrid Cluster with Torus Interconnect Aimed at Molecular-Dynamics Simulations

Vladimir Stegailov^{1,2,3} , Alexander Agarkov⁴, Sergey Biryukov⁴, Timur Ismagilov⁴, Mikhail Khalilov³, Nikolay Kondratyuk^{1,2,3} , Evgeny Kushtanov⁴, Dmitry Makagon⁴, Anatoly Mukosey⁴, Alexander Semenov⁴, Alexey Simonov⁴, Alexey Timofeev^{1,2,3} , and Vyacheslav Vecher^{1,2}

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

² Moscow Institute of Physics and Technology, Dolgoprudny, Russia

³ National Research University Higher School of Economics, Moscow, Russia

v.stegailov@hse.ru

⁴ JSC NICEVT, Moscow, Russia

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 GRO-MACS, LAMMPS and VASP and compare the results with other HPC systems. This cluster serves as a test bed for the Angara interconnect that supports 3D and 4D torus network topologies, and verifies its ability to unite MPP systems speeding-up effectively MPI-based applications. We describe the interconnect presenting typical MPI benchmarks.

Keywords: Atomistic simulations · Angara interconnect · GPU

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 fs).

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 (e.g. [4]).

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 that boosts 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 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 [5–7]. The sustained performance is usually limited by the memory-bound nature of the algorithms.

Among GPU-aware MD software one can point out GROMACS [8] as, perhaps, the most computationally efficient MD tool and LAMMPS [9] as one of the most versatile and flexible for MD models creation. Different GPU off-loading schemes were implemented in LAMMPS [10–13]. GROMACS provides a highly optimized GPU-scheme as well [14].

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 [15] or ANTON [3]. Currently, general purpose Nvidia GPUs provide the most cost-effective way for MD calculations [16].

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 interconnect (for the first time in English) and the resulting performance of the cluster for MD models in GROMACS and LAMMPS and for the DFT calculations in VASP.

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 [17] that 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 [18]. 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 [19].

The AURORA Booster and GREEN ICE Booster supercomputers are based on the EXTOLL torus interconnect [20].

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 [21], the FPGA prototypes of the SKIF-Aurora [22] and Pautina [23] torus interconnects. Up to this moment the Angara interconnect has been evolving all the way from the FPGA prototype [24, 25] to the ASIC-based card [26].

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 [27]. 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 [6].

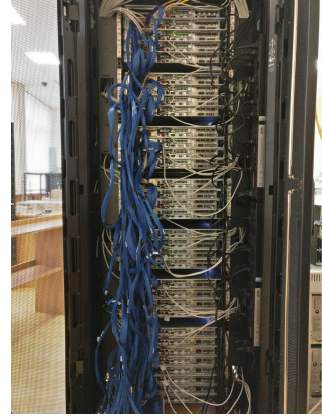


Fig. 1. The Desmos cluster.

3 The Desmos Cluster

The hardware was selected in order to maximize the number of nodes and the efficiency of a one node for MD workloads. Each node consists of Supermicro SuperServer 1018GR-T, Intel Xeon E5-1650v3 (6 cores, 3.5 GHz) and Nvidia GeForce 1070 (8 GB GDDR5) and has DDR4-2133 (16 GB).

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 sure that there is no hardware memory errors in each GPU. Testing of each GPU was performed using MemtestG80 [28] during more than 4 h 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 8 GB 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 $4 \times 2 \times 2 \times 2$, copper Samtec cables). 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 GigE 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.

Table 1. 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

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 [20]. The Angara chip supports deadlock-free adaptive routing based on bubble flow control [29], direction ordered routing [17,18] and initial and final hops for fault tolerance [17].

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

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 1 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 μ s for the 16 bytes message size and exceeds 4x FDR interconnect for all small message sizes.

We use Intel MPI Benchmarks to evaluate MPIBarrier and MPIAlltoall operation times for different number of nodes of the Desmos and Polyteknik clusters. The Angara superiority for small messages explains better results for MPIBarrier (Fig. 2b) and MPIAlltoall with 16 byte messages (Fig. 2c). For large messages (256 Kbytes) the Desmos results are worse than that of the Polyteknik cluster (Fig. 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.

We have considered a heuristic algorithm to optimize topology-aware mapping of MPI-processes on the physical topology of the Desmos cluster. The algorithm distributes processes on CPU cores optimally to minimize exchange times [30]. Preliminary experimental results for NAS Parallel Benchmarks show about 50% performance improvements.

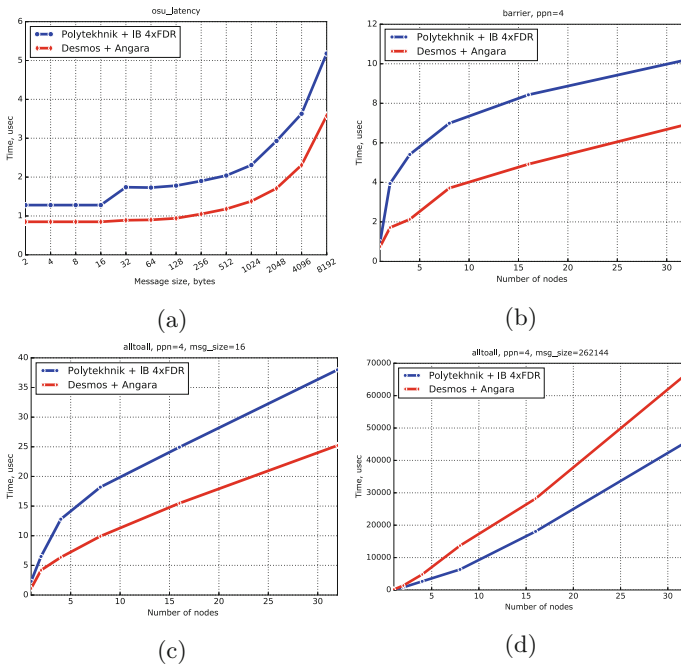


Fig. 2. The OSU latency between two adjacent nodes (a) Times for MPIBarrier with 4 processes per node (b) Times for MPIAlltoall with 4 processes per node and message sizes 16 b (c) and 256 Kb (d).

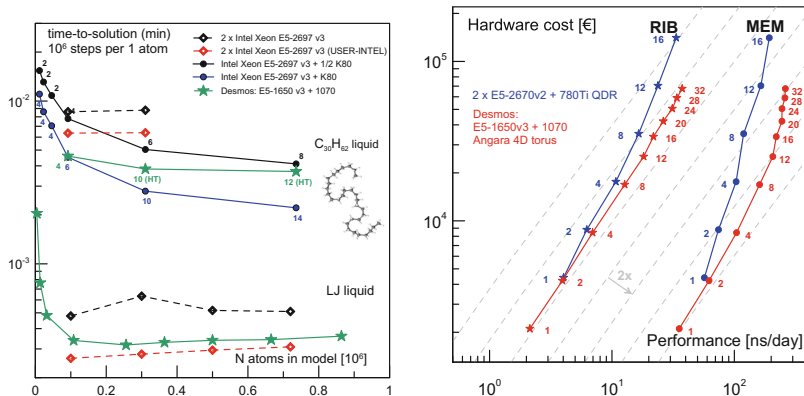


Fig. 3. Left: time per atom per timestep for LJ liquid and C₃₀H₆₂ oil LAMMPS MD models of different size and different hardware combinations (the numbers near symbols show the number of MPI threads used). Right: the hardware cost vs achieved performance for MEM and RIB benchmarks. The dashed lines show ideal scaling. The Desmos results are compared with the published data [16]. (Color figure online)

5 Classical 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 confirmed the broad applicability of this approach. Recent developments of optimized MD algorithms include the validation of the single precision solver (e.g. [31]). 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 Fig. 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 C₃₀H₆₂ 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.

The work [16] 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, ~100 k atoms) and the

bacterial ribosome in water with ions (RIB, ~ 2 M atoms). The GROMACS package was used for all tests.

We compare the results obtained on Desmos cluster with the best choice of [16]: the nodes that consist of 2 socket Xeon E5-2670 v2 with 2 780Ti and connected via IB QDR. The costs of hardware in Euros are displayed on Y axis (excluding the cost of the interconnect as in [16]) and the performance in ns/day is shown on X axis in Fig. 3. The numbers show the number of nodes used.

Desmos (red color) shows better strong scaling for the MEM benchmark than the best system configuration provided by Kutzner et al. (blue color) [16]. 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.

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

6 Benchmarks with the Electronic Structure Code VASP

The main computing power of Desmos cluster consists in GPU and is aimed at classical MD calculations. Each node has only one 6 core processor. However, it is interesting methodically to produce scaling tests for DFT calculations of electronic structure. DFT calculations are highly dependent on the speed of collective all-to-all exchanges in contrast to classical MD models. We use one of the most used packages VASP. According to the current estimates [32, 33], the calculations carried out in VASP package consume up to 20% of the whole computing time in the world.

The GaAs crystal model consisted of 80 atoms is used for the test in VASP [34]. The results of tests are shown on Fig. 4. The time for 1 iteration of self-consistent electronic density calculation is shown depending on the peak performance. The data for different clusters are presented: MVS10P and MVS1P5 supercomputers of Joint Supercomputer Centre of Russian Academy of Sciences and Boreasz IBM 775 of Warsaw University. The numbers near the symbols are the numbers of nodes for each test. The lower of two points showing the computing time on 32 Desmos nodes corresponds to additional parallelization over k-points.

The obtained results show that the Angara network very effectively unites the cluster nodes together. The supercomputers that are used for comparison

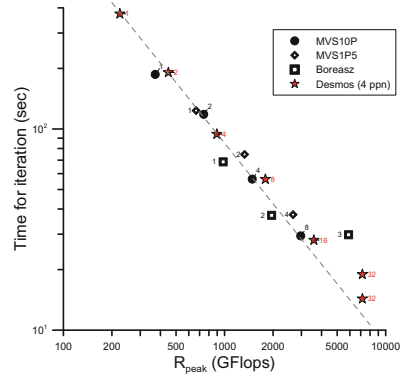


Fig. 4. The scaling of the GaAs crystal model in VASP.

have two-socket nodes (IBM 775 has even four-socket nodes). Nevertheless, MPI-exchanges over the Angara network in terms of resulting performance give the same result as MPI-exchanges in shared memory.

7 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 scaling tests for the electronic structure calculations also show the high efficiency of the MPI-exchanges over the Angara network. The Desmos cluster is the first application of the Angara interconnect for a GPU-based MPP system. The features of the Angara interconnect provided the high 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. The authors acknowledge Joint Supercomputer Centre of Russian Academy of Sciences for the access to the MVS-10P supercomputer.

Acknowledgments. The JIHT team was supported by the Russian Science Foundation (grant No. 14-50-00124). Their work included the development of the Desmos cluster architecture, tuning of the codes and benchmarking (HSE and MIPT provided preliminary support). The NICEVT team developed the Angara interconnect and its low-level software stack, built and tuned the Desmos cluster.

The authors are grateful to Dr. Maciej Cytowski and Dr. Jacek Peichota (ICM, University of Warsaw) for the data on the VASP benchmark [34].

References

1. Heinecke, A., Eckhardt, W., Horsch, M., Bungartz, H.-J.: Supercomputing for Molecular Dynamics Simulations. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-319-17148-7>
2. Eckhardt, W., Heinecke, A., Bader, R., Brehm, M., Hammer, N., Huber, H., Kleinhenz, H.-G., Vrabec, J., Hasse, H., Horsch, M., Bernreuther, M., Glass, C.W., Niethammer, C., Bode, A., Bungartz, H.-J.: 591 TFLOPS multi-trillion particles simulation on SuperMUC. In: Kunkel, J.M., Ludwig, T., Meuer, H.W. (eds.) ISC 2013. LNCS, vol. 7905, pp. 1–12. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38750-0_1
3. Piana, S., Klepeis, J.L., Shaw, D.E.: Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Curr. Opin. Struct. Biol.* **24**, 98–105 (2014)
4. Begau, C., Sutmann, G.: Adaptive dynamic load-balancing with irregular domain decomposition for particle simulations. *Comput. Phys. Commun.* **190**, 51–61 (2015)
5. Smirnov, G.S., Stegailov, V.V.: Efficiency of classical molecular dynamics algorithms on supercomputers. *Math. Models Comput. Simul.* **8**(6), 734–743 (2016)

6. Stegailov, V.V., Orekhov, N.D., Smirnov, G.S.: HPC hardware efficiency for quantum and classical molecular dynamics. In: Malyshkin, V. (ed.) PaCT 2015. LNCS, vol. 9251, pp. 469–473. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21909-7_45
7. Rojek, K., Wyrzykowski, R., Kuczynski, L.: Systematic adaptation of stencil-based 3D MPDATA to GPU architectures. *Concurr. Comput. Pract. Exp.* **29**, e3970 (2016)
8. Berendsen, H.J.C., van der Spoel, D., van Drunen, R.: Gromacs: a message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.* **91**(13), 43–56 (1995)
9. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**(1), 1–19 (1995)
10. Trott, C.R., Winterfeld, L., Crozier, P.S.: General-purpose molecular dynamics simulations on GPU-based clusters. ArXiv e-prints (2010)
11. Brown, W.M., Wang, P., Plimpton, S.J., Tharrington, A.N.: Implementing molecular dynamics on hybrid high performance computers - short range forces. *Comput. Phys. Commun.* **182**(4), 898–911 (2011)
12. Brown, W.M., Wang, P., Plimpton, S.J., Tharrington, A.N.: Implementing molecular dynamics on hybrid high performance computers - Particle-particle particle-mesh. *Comput. Phys. Commun.* **183**(3), 449–459 (2012)
13. Edwards, H.C., Trott, C.R., Sunderland, D.: Kokkos: enabling manycore performance portability through polymorphic memory access patterns. *J. Parallel Distrib. Comput.* **74**(12), 3202–3216 (2014). Domain-specific languages and high-level frameworks for high-performance computing
14. Abraham, M.J., Murtola, T., Schulz, R., Páll, S., Smith, J.C., Hess, B., Lindahl, E.: Gromacs: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **12**, 19–25 (2015)
15. Ohmura, I., Morimoto, G., Ohno, Y., Hasegawa, A., Taiji, M.: MDGRAPE-4: a special-purpose computer system for molecular dynamics simulations. *Philos. Trans. R. Soc. Lond. Math. Phys. Eng. Sci.* **372**, 2014 (2021)
16. Kutzner, C., Pall, S., Fechner, M., Esztermann, A., de Groot, B.L., Grubmüller, H.: Best bang for your buck: GPU nodes for GROMACS biomolecular simulations. *J. Comput. Chem.* **36**(26), 1990–2008 (2015)
17. Scott, S.L., Thorson, G.M.: The Cray T3E network: adaptive routing in a high performance 3D torus. In: HOT Interconnects IV, Stanford University, 15–16 Aug 1996
18. Adiga, N.R., Blumrich, M.A., Chen, D., Coteus, P., Gara, A., Giampapa, M.E., Heidelberger, P., Singh, S., Steinmacher-Burow, B.D., Takken, T., Tsao, M., Vranas, P.: Blue Gene/L torus interconnection network. *IBM J. Res. Dev.* **49**(2), 265–276 (2005)
19. Ajima, Y., Inoue, T., Hiramoto, S., Takagi, Y., Shimizu, T.: The Tofu interconnect. *IEEE Micro* **32**(1), 21–31 (2012)
20. Neuwirth, S., Frey, D., Nuessle, M., Bruening, U.: Scalable communication architecture for network-attached accelerators. In: 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), pp. 627–638, February 2015
21. Elizarov, G.S., Gorbunov, V.S., Levin, V.K., Latsis, A.O., Korneev, V.V., Sokolov, A.A., Andryushin, D.V., Klimov, Y.A.: Communication fabric MVS-Express. *Vychisl. Metody Programm.* **13**(3), 103–109 (2012)

22. Adamovich, I.A., Klimov, A.V., Klimov, Y.A., Orlov, A.Y., Shvorin, A.B.: Thoughts on the development of SKIF-Aurora supercomputer interconnect. *Programmnyye Sistemy: Teoriya i Prilozheniya* **1**(3), 107–123 (2010)
23. Klimov, Y.A., Shvorin, A.B., Khrenov, A.Y., Adamovich, I.A., Orlov, A.Y., Abramov, S.M., Shevchuk, Y.V., Ponomarev, A.Y.: Pautina: the high performance interconnect. *Programmnyye Sistemy: Teoriya i Prilozheniya* **6**(1), 109–120 (2015)
24. Korzh, A.A., Makagon, D.V., Borodin, A.A., Zhabin, I.A., Kushtanov, E.R., Syromyatnikov, E.L., Cheryomushkina, E.V.: Russian 3D-torus interconnect with globally addressable memory support. *Vestnik YuUrGU. Ser. Mat. Model. Progr.* **6**, 41–53 (2010)
25. Mukosey, A.V., Semenov, A.S., Simonov, A.S.: Simulation of collective operations hardware support for Angara interconnect. *Vestn. YuUrGU. Ser. Vych. Mat. Inf.* **4**(3), 40–55 (2015)
26. Agarkov, A.A., Ismagilov, T.F., Makagon, D.V., Semenov, A.S., Simonov, A.S.: Performance evaluation of the Angara interconnect. In: *Proceedings of the International Conference “Russian Supercomputing Days” – 2016*, pp. 626–639 (2016)
27. Corsetti, F.: Performance analysis of electronic structure codes on HPC systems: a case study of SIESTA. *PLoS ONE* **9**(4), 1–8 (2014)
28. Haque, I.S., Pande, V.S.: 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 2010*, pp. 691–696. IEEE Computer Society, Washington (2010)
29. Puente, V., Beivide, R., Gregorio, J.A., Prellezo, J.M., Duato, J., Izu, C.: Adaptive bubble router: a design to improve performance in torus networks. In: *Proceedings of the 1999 International Conference on Parallel Processing*, pp. 58–67 (1999)
30. Hoefler, T., Snir, M.: Generic topology mapping strategies for large-scale parallel architectures. In: *Proceedings of the International Conference on Supercomputing, ICS 2011*, pp. 75–84. ACM, New York (2011)
31. Höhnerbach, M., Ismail, A.E., Bientinesi, P.: 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 2016*, pp. 7:1–7:13. IEEE Press, Piscataway (2016)
32. Bethune, I.: *Ab Initio Molecular Dynamics. Introduction to Molecular Dynamics on ARCHER* (2015)
33. Max Hutchinson. VASP on GPUs. When and how. In: *GPU Technology Theater, SC15* (2015)
34. Cytowski, M.: Best practice guide – IBM power 775. In: *PRACE* (2013)