

Research Article

Igor Zacharov*, Rinat Arslanov, Maksim Gunin, Daniil Stefonishin, Andrey Bykov, Sergey Pavlov, Oleg Panarin, Anton Maliutin, Sergey Rykovanov, and Maxim Fedorov

“Zhores” — Petaflops supercomputer for data-driven modeling, machine learning and artificial intelligence installed in Skolkovo Institute of Science and Technology

<https://doi.org/10.1515/eng-2019-0059>

Received May 04, 2019; accepted Aug 28, 2019

Abstract: The Petaflops supercomputer “Zhores” recently launched in the “Center for Computational and Data-Intensive Science and Engineering” (CDISE) of Skolkovo Institute of Science and Technology (Skoltech) opens up new exciting opportunities for scientific discoveries in the institute especially in the areas of data-driven modeling, machine learning and artificial intelligence. This supercomputer utilizes the latest generation of Intel and NVidia processors to provide resources for the most compute intensive tasks of the Skoltech scientists working in digital pharma, predictive analytics, photonics, material science, image processing, plasma physics and many more. Currently it places 7th in the Russian and CIS TOP-50 (2019) supercomputer list. In this article we summarize the cluster properties and discuss the measured performance and usage modes of this new scientific instrument in Skoltech.

Keywords: High Performance Computing, High Speed Networks, Parallel Computation, Computing Clusters, Energy Efficiency, Computer Scalability

1 Introduction

Modern science, industry and business benefit greatly from using high-performance computing (HPC). In the recent years there is a clear trend for convergence of traditional HPC, Machine Learning, Data Science and Artificial Intelligence [24, 26, 30, 32, 34]. Additionally, exponentially growing amount of both structured and unstructured data obtained from various sources, including, but not limited to Internet of Things (IoT) and mathematical modeling, led to the appearance of the notion of data-intensive science being the fourth paradigm of science [10] along with experiments, theory and computer simulations. Indeed, one can gain a lot of new knowledge about the universe by processing the data, which is hardly useful otherwise. One can also see a trend for a multidisciplinary approach to the traditionally “computational” problems. For example, deep learning can outperform density functional theory (DFT) in quantum chemistry [28], and in solving ordinary and partial differential equations [12, 14]. Skoltech has designed this new supercomputer in view of these trends and we report on the architecture and new research areas enabled by it in this article.

Skoltech CDISE Petaflops supercomputer “Zhores” named after the Nobel Laureate Zhores Alferov is intended for cutting-edge multidisciplinary research in data-driven simulations and modeling, machine learning, Big Data and artificial intelligence (AI). It enables research in such important fields as Bio-medicine, Computer Vision, Remote Sensing and Data Processing, Oil/Gas, Internet of Things (IoT), High Performance Computing [9, 21], Quantum Computing, Agro-informatics, Chemical-informatics, development of novel X- and gamma-ray sources [27] (which was the first published work that used the “Zhores” supercomputer) and many more. Its architecture reflects the modern trend of convergence of “traditional” HPC, Big Data and AI. Moreover, heterogeneous

*Corresponding Author: Igor Zacharov: Skolkovo Institute of Science and Technology (Skoltech), cluster for Computational and Data-Intensive Science and Engineering (CDISE), 143026 Moscow, Russian Federation; Email: i.zacharov@skoltech.ru

Rinat Arslanov, Maksim Gunin, Daniil Stefonishin, Andrey Bykov, Sergey Pavlov, Oleg Panarin, Anton Maliutin, Sergey Rykovanov, Maxim Fedorov: Skolkovo Institute of Science and Technology (Skoltech), cluster for Computational and Data-Intensive Science and Engineering (CDISE), 143026 Moscow, Russian Federation

demands of Skoltech projects on computing possibilities ranging from throughput computing to capability computing and the need to apply modern concepts of workflow acceleration and in-situ data analysis impose corresponding solutions on the architecture. The design of the cluster is based on the latest generation of CPUs, GPUs, network and storage technologies, current as of 2017–2019. This paper describes the implementation of this machine and gives details of the initial benchmarks that validate its architectural concepts.

The article is organized as follows. In section 2 the details of installation are discussed with subsections dedicated to the basic technologies. Section 3 describes several applications ran on the “Zhores” cluster and their scaling. The usage of the machine in the “Neurohackaton” held in November 2018 in Skoltech is described in section 4. Finally, section 5 provides conclusions.

2 Installation

“Zhores” is constructed from the DELL PowerEdge C6400 and C4140 servers with Intel® Xeon® CPUs and Nvidia Volta GPUs connected by Mellanox EDR Infiniband (IB) SB7800/7890 switches. We decided to allocate 20 TB of the fastest storage system (based on NVMe over IB technology) for small users’ files and software (home directories), and 0.6 PB GPFS file system for bulk data storage. The principal scheme with the majority of components is illustrated in Figure 1. The exact composition with the characteristics of the components is found in Table 1. The names of the nodes are given according to their intended role:

- cn — compute nodes to handle CPU workload
- gn — compute nodes to handle GPU workload
- hd — hadoop nodes with set of disks for the classical Hadoop workload
- an — access nodes for cluster login, submit jobs and transfer users’ data
- anlab — special nodes for user experiments
- vn — visualization nodes
- mn — main nodes for cluster management and monitoring

All users land on one of the access nodes (an) after login and can use them for interactive work, data transfer and for job submission (dispatching tasks to compute nodes). Security requirements place the access nodes in the demilitarized zone. The queue structure is implemented using the SLURM workload manager and discussed in section 2.5. Both, shell scripts and Docker [4]

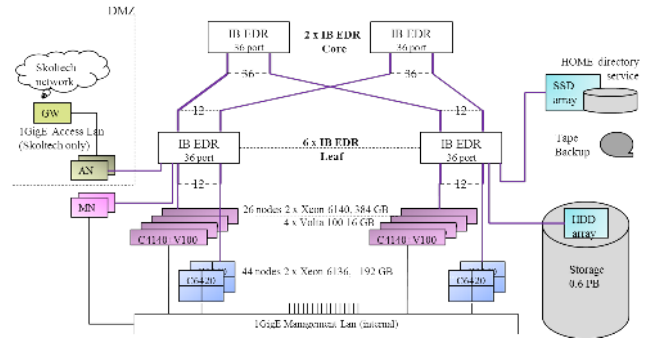


Figure 1: Principle connection scheme. The an and mn nodes are marked explicitly; the cn, gn and other nodes are lumped together.

images are accepted as valid work item by the queuing system. We have made a principal decision to use the latest CentOS version 7.5 which was officially available at the time of installation. The user environment is provided with the Environment Modules software system [5]. Several compilers (Intel and GNU) are available as well as different versions of pre-compiled utilities and applications.

The cluster is managed with the fault tolerant installation of the Luna management tool [8]. The two management nodes are mirrors of each other and provide the means of provisioning and administration of the cluster, provide the NFS export of user/home directories and all cluster configuration data. This is described in section 2.4.

2.1 Servers’ Processor Characteristics

The servers have the latest generation of the Intel Xeon processors and Nvidia Volta GPUs. The basic characteristics of each type of the servers are captured in Table 1. We have measured the salient features of these devices.

Intel Xeon 6136 and 6140 “Gold” CPUs of Skylake generation differ by the total number of cores in the package and the working clock frequency (F). Each core features two floating point AVX512 units. This has been tested with a special benchmark to verify that the performance varies with the frequency as expected.

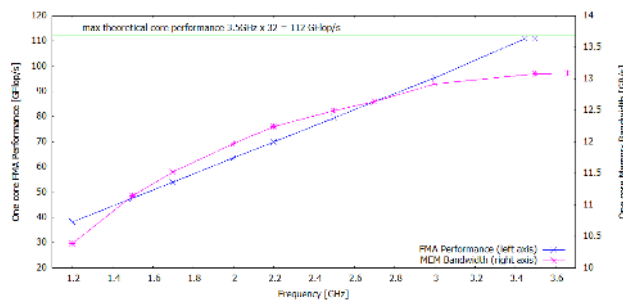
The CPU performance and memory bandwidth of a single core is shown in Figure 2. The benchmark program to test the floating point calculation performance is published elsewhere [3]. It is an unrolled vector loop with vector width 8, precisely tuned for the AVX512 instruction set. In this loop exactly 8 double precision numbers will be computed in parallel in two execution units of each core. With two execution units and the fused multiply-add instruction (FMA) the theoretical Double Precision

Table 1: Details of named “Zhores” cluster nodes

Name	CPU	sockets × cores	F [GHz]	Memory [GB]	Storage [TB]	[TF/s]	#	[TF/s]
cn	6136	2 × 12	3.0	192	0.48	2.3	44	101.4
gn	6140	2 × 18	2.3	384	0.48	2.6	26	68.9
	V100	4 × 5120	1.52	4 × 16		31.2		811.2
hd	6136	2 × 12	3.0	192	9.0	2.3	4	9.2
an	6136	2 × 12	3.0	256	4.8	2.3	2	4.6
vn	6134	2 × 8	3.2	384		1.6	2	3.2
anlab	6134	2 × 8	3.2	192		3.3	4	13.1
mn	6134	2 × 8	3.2	64		3.3	2	6.6
Totals		2296		21248			82	1018.2

Table 2: Memory properties from Xeon 6136/6140 processor visible from single core

cache level	set	line [Bytes]	Latency [ns]	Bandwidth [GB/s]	size [KiB]	Core OWN
L1 Data	8-way	64	1.1	58	32	private
L1 Instr.	8-way	64			32	private
L2 Unif.	16-way	64	3.8	37	1024	private
L3 Unif.	11-way	64		26	25344	shared
TLB	4-way				64 entries	private
Memory Xeon 6136 parts			27.4	13.1	192 GB	shared
Memory Xeon 6140 parts			27.4	13.1	384 GB	shared

**Figure 2:** Floating point performance (FMA instructions) on 6136 CPU core and memory bandwidth (STREAM Triad) as a function of clock frequency. Left ordinate shows the FMA performance, the right ordinate represents the memory bandwidth.

(DP) performance of a single physical core is $8 \times 2 \times 2 \times F$ [GHz] and for the maximum of $F = 3.5$ GHz may reach 112 GFlop/s/core. The performance scales with the frequency to the maximum determined by processor thermal and electrical limits. The total FMA performance on a node when running AVX512 code on all processors in parallel is about 2.0 TFlop/s for C6140 machines (cn nodes, 24 cores) and 2.4 TFlop/s for the C4140 (gn nodes, 36 cores). Summing up all the cn and gn nodes gives the measured maximum CPU performance on the “Zhores” cluster of 150 TFlop/s.

The latencies of the processor memory subsystem have been measured with the LMBench program [23] and summarized in Table 2.

The main memory performance is measured with the STREAMS program [22] and shown for the single core as a function of clock frequency in Figure 2. The theoretical performance of the memory bandwidth may be estimated with the Little Law [15] to 14 GB/s per each channel taking into account the memory latency of 27.4 ns given in Table 2. The total memory bandwidth (STREAM Triad) for all cores reached 178.6 GB/s in our measurement using all 6 channels of 2666 MHz DIMMs.

The strong dependency of the FMA performance on the processor clock frequency and the weak dependency of the memory bandwidth on the clock frequency is noted to propose a scheme for the optimization of the power usage for applications with mixed instruction profiles.

2.2 Nvidia V100 GPU

Significant nodes (26) in the “Zhores” cluster are equipped with four Nvidia V100 GPUs each. The GPUs are connected pairwise with NVLink and individually with PCIe gen3 x16 to the CPU host. The principal scheme of the connections is shown in Figure 3. The basic measurements to label the

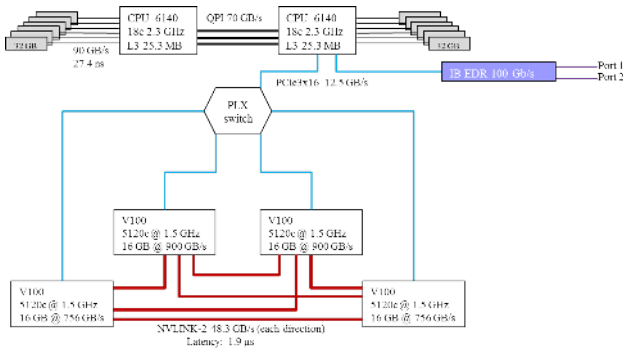


Figure 3: Principal connections between host and graphics subsystem on graphics nodes.

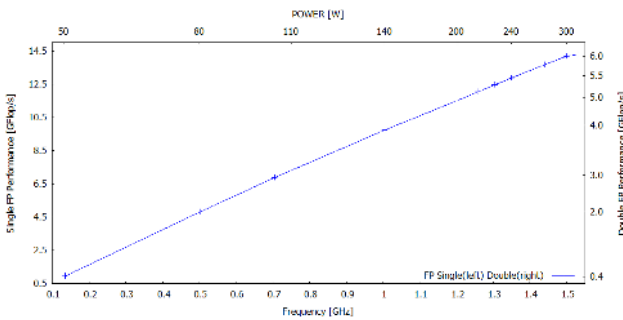


Figure 4: Nvidia V100 GPU floating point performance as a function of graphics clock rate. Electrical power draw corresponding to the set frequency is indicated on the upper axis.

links in the plot have been obtained with Nvidia p2p bandwidth program from the “Samples” directory loaded with GPU drivers. This setup is optimized for parallel computation scaling within the node, while the connections to the cluster network pass from the single PCIe link. The maximum estimated performance of a single V100 GPU is shown in Figure 4. The graphics clock rate was set with the command “nvidia-smi”; same command with different parameters lists the power draw of the device. The computational efficiency measured in Performance per Watt is not evenly distributed as function of frequency, the peak is 67.4 GFlop/s/W (single precision) at 1 GHz and drops to 47.7 GFlop/s/W at 1.5 GHz.

2.3 Mellanox IB EDR network

The high performance cluster network has the Fat Tree topology and is build from six Mellanox SB7890 (unmanaged) and two SB7800 (managed) switches that provide 100 Gbit/s (IB EDR) connections between the nodes. The performance of the interconnect has been measured with the “mpilink” program that times the ping-pong exchange

between each node [1]. To make the measurements we have installed Mellanox HPC package drivers and used openMPI version 3.1.2. The results are shown in Figure 5 for serial mode runs and in Figure 6 for parallel mode runs.

The serial mode sends packages to each node when previous communication has finished, while in parallel mode all sends and receives are issued at the same time. The parallel mode probes the package contention, while serial mode allows to establish the absolute speed and discover any failing links. The communication in serial mode is centered around the speed of $10.2 \pm 0.5 \text{ GB/s}$. The parallel mode reveals certain over-subscription of the Fat Tree network — while the computational nodes are balanced the additional traffic from the file services causes delays in the transmission. This problem will be addressed in future upgrades.

2.4 Operating System and cluster management

The “Zhores” cluster is managed by “Luna” [8] provisioning tool which can be installed in a fault tolerant active-passive cluster setup with TrinityX platform. The Luna management system was developed by ClusterVision BV. The system automates the creation of all the services and cluster configuration that make a bunch of servers a unified computational machine.

The cluster management software supports the following essential features:

- All cluster configuration is kept in the Luna database and all cluster nodes boot from this information which is held in one place. This database is mirrored between the management nodes with the DRBD filesystem and the active management node provides access to data for every node in the cluster with the NFS share, see Figure 7.
- Node provisioning from OS images is based on the BitTorrent protocol [2] for efficient simultaneous (disk-less or standard) boot; the image management allows to grab an OS image from a running node to a file, clone images for testing or backup purposes; a group of nodes can use the same image for provisioning that fosters unification of cluster configuration. Nodes use PXE protocol to load a service image that implements the booting procedure.
- All the nodes (or groups of nodes) in the cluster can be switched on/off and reset with the IPMI proto-



Figure 5: Histogram of the ping-ping times/speeds between all nodes using 1 MB packets in serial mode

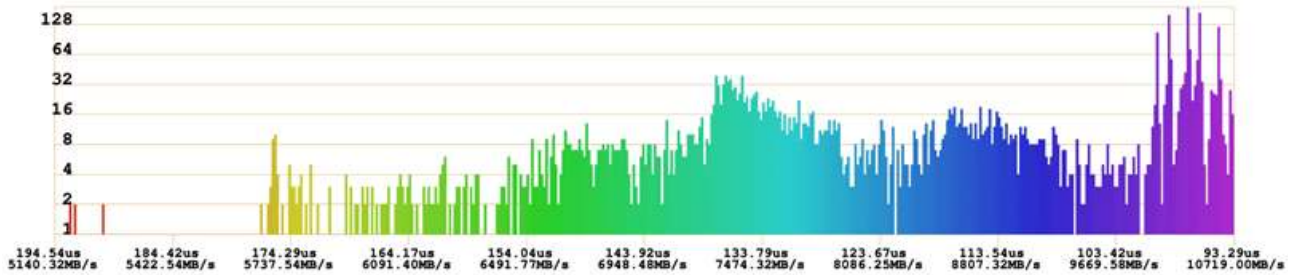


Figure 6: Histogram of the ping-ping times/speeds between all nodes using 1 MB packets in parallel mode

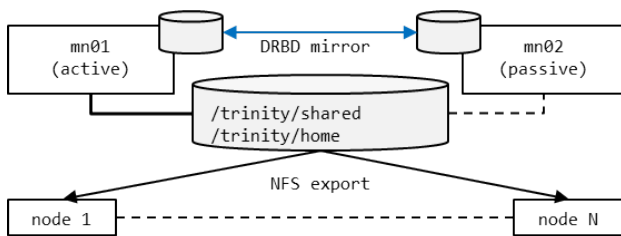


Figure 7: Organization of the “Zhores” cluster management with Luna system.

col from the management nodes with a single command.

- The cluster services setup on the management node in a fault tolerant way include the following: DHCP, DNS, OpenLDAP, Slurm, Zabbix, Docker-repository, etc.

The management nodes are based on CentOS 7.5 and force same OS on the compute nodes; additional packages, specific drivers and different kernel versions can be included in the images for the cluster nodes. The installation requires each node to have at least two ethernet network interfaces, one dedicated to the management traffic and the other used for administrative access. A single cluster node can be booted within 2.5 minutes (over 1 GbE), and the whole “Zhores” cluster cold start takes 5 minutes to fully operational state.

2.5 The queuing system

Work queues have been organized with the Slurm workload manager to reflect the different application profiles of users of the cluster. Several nodes have been given to dedicated projects (gn26, anlab) and one CPU-only node is setup for debugging work (cn44). The remaining nodes have been combined in queues for the GPU-nodes (gn01–gn25) and for the CPU-nodes (cn01–cn43).

2.6 Linpack run

The Linpack benchmark was performed as a part of the cluster evaluation procedure and to rate the supercomputer for the performance comparison. The results of the run are shown in Table 3 separately for the GPU and all nodes using only CPU computation.

The “Zhores” supercomputer is significant for the Russian computational science community, and currently, it places 7th in the Russian and CIS TOP-50 list [6].

Table 3: Linpack performance of the “Zhores” cluster run separately on the GPU nodes and with all CPU resources. The power draw for CPU Linpack run is estimated (*).

Part nodes/core	N	NB P Q	T [s]	R_{max} [TFlop/s]	R_{peak} [TFlop/s]	eff. [%]	P [kW]
gn01-26 26/930	452352	192 13 8	124.4	496($\pm 2\%$)	811.2	61.1	48.9
gn;cn;hn 72/2028	1125888	384 12 12	7913.5	120.2($\pm 2\%$)	158.7	75.6	35*

Table 4: Computational times for 16 time-integration steps for the parallel implementation of algorithm for the aggregation and fragmentation equations with $N = 2^{22}$ strongly-coupled nonlinear ODEs. In this benchmark we utilized the nodes from the CPU segment of the cluster.

Number of CPU cores	Time, sec
1	585.90
2	291.69
4	152.60
8	75.60
16	41.51
32	20.34
64	12.02
128	6.84

Table 5: Parallel advection-coagulation solver on CPUs, Ballistic kernel, domain size $N \times M = 12288$, 16 time-integration steps. This benchmark utilized up to 32 nodes from the CPU segment of the cluster.

Number of CPU cores	Time, sec
1	1706.50
2	856.057
4	354.85
8	224.44
12	142.66
16	105.83
24	79.38
48	38.58
96	19.31
192	9.75
384	5.45
768	4.50

3 Applications

3.1 Algorithms for aggregation and fragmentation equations

In our benchmarks we used parallel implementation of efficient numerical methods for the aggregation and fragmentation equations [13, 18] and also parallel implementation of the solver for advection-driven coagulation process [16]. Its sequential version has already been utilized in a number of applications [17, 19] and can be considered as one of the most efficient algorithms for a class of Smoluchowski-type aggregation kinetic equations. It is worth to stress that parallel algorithm for pure aggregation-fragmentation equations is based mostly on the performance of ClusterFFT operation which is a dominating operation in terms of algorithmic complexity, thus its scalability is extremely limited. Nevertheless for 128 cores we obtain speedup of calculations by more than 85 times, see Table 4.

In the case of the parallel solver, for advection-driven coagulation [21] we obtain almost ideal acceleration with utilization of the algorithm for almost full CPU-based segment. In this case, the algorithm is based on the one-dimensional domain decomposition along the spatial coordinate and has a very good scalability, see Table 5 and Figure 8. The experiments have been performed using Intel® compilers and the Intel® MKL library.

Alongside with the consideration of the well-known two-particle problem of aggregation, we have measured the performance for a parallel implementation of a more general three-particle (ternary) Smoluchowski-type kinetic aggregation equations [29]. In this case the algorithm is somewhat similar to the one for standard binary aggregation. However the number of the floating point calculations and the size of the allocated memory increases as compared to the binary case, because the dimension of the low rank Tensor Train (TT) decomposition [25] is naturally bigger in ternary case. The most computationally expensive operation in the parallel implementation of the al-

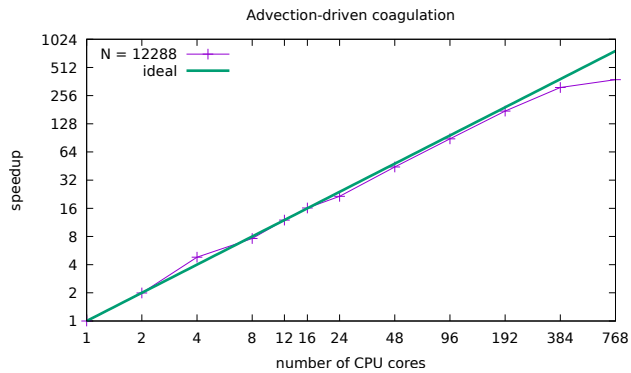


Figure 8: Parallel advection-driven aggregation solver on CPUs, Ballistic kernel, domain size $N = 12288$.

Table 6: Computational times for 10 time-integration steps for parallel implementation of the algorithm for ternary aggregation equations with $N = 2^{19}$ nonlinear ODEs.

Number of CPU cores	Time, sec
1	624.19
2	351.21
4	186.83
8	100.33
16	52.02
32	33.74
64	27.74
128	24.80

gorithm is also the ClusterFFT. The speedup of the parallel ternary aggregation algorithm applied to the empirically derived ballistic-like kinetic coefficients [20] is shown in Table 6. In full accordance with the structure of ClusterFFT and the problem complexity one needs to increase the parameter N of the used differential equations in order to obtain scalability. Speedups for both implementations of binary and ternary aggregation are shown on Figure 9. The experiments have been performed using Intel compilers and Intel MKL library.

3.2 Gromacs

Classical molecular dynamics is an effective method with high predictive ability in a wide range of scientific fields [11, 31]. Using Gromacs 2018.3 software [7, 33] we have performed molecular dynamics simulations in order to test the “Zhores” cluster performance. As a model system we chose 125 million Lennard-Jones spheres with the Van der Waals cut-off radius of 1.2 nm and with the Berendsen

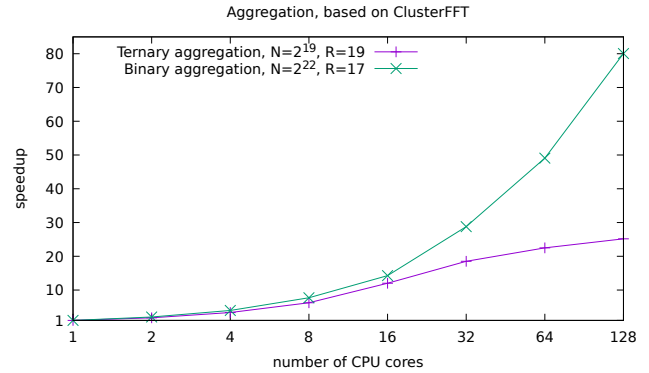


Figure 9: Parallel binary and ternary aggregation solvers on CPU, Ballistic-like kernels, 16 and 10 time-integration steps for $N = 2^{22}$ and $N = 2^{19}$ nonlinear ODEs, respectively. Parameter R denotes the rank of used matrix and tensor decompositions.

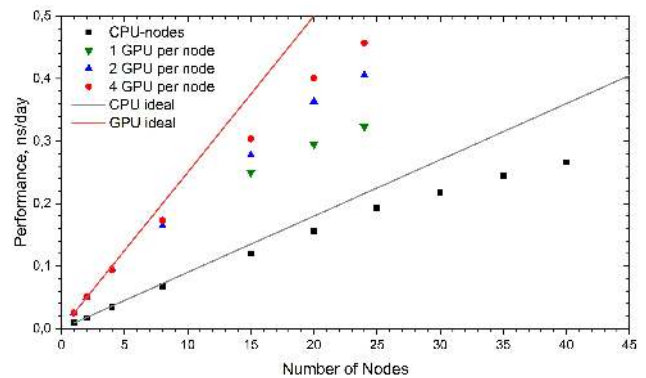


Figure 10: Performance of the molecular dynamic simulations of 125 million Lennard-Jones spheres using Gromacs 2018.3 as a function of nodes number. Note, that there are only 26 GPU nodes in the cluster.

thermostat. All tests were conducted with a single precision version of Gromacs.

The results are presented in Figure 10. We measured the performance as a function of the number of nodes; we have used up to 40 CPU nodes and up to 24 GPU nodes. We have used 4 OpenMP threads per MPI process. Each task was performed 5 times with following averaging in order to obtain final performance. Grey and red solid lines show linear acceleration of the program on CPU and GPU nodes, respectively. In case of the CPU-nodes, one can see almost ideal speedup. With a large number of CPU-nodes, the speedup deviates from linear and grows slower.

To test performance on the GPU-nodes, we have performed simulations with 1, 2 and 4 graphics cards per node. The use of all 4 graphics cards demonstrates good scalability, while 2 GPUs per node shows slightly lower speedup. Runs with 1 GPU per node demonstrates worse performance, especially with high number of nodes. To compare

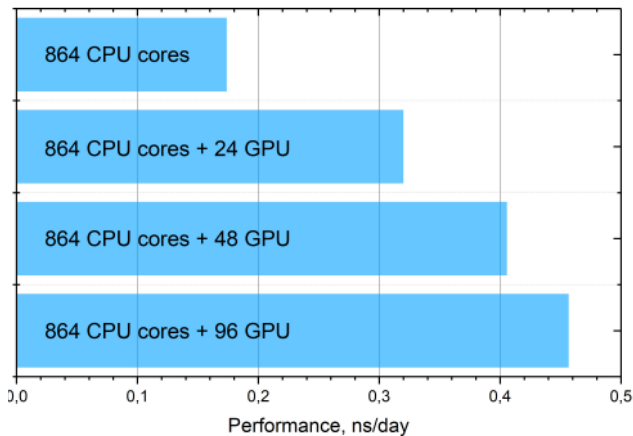


Figure 11: Performances for different configurations of 24 GPU-nodes: 0, 1, 2, and 4 GPUs per node

the efficiency for different number of GPUs per node, we show the performance for the four configurations (0, 1, 2 and 4 GPUs) using 24 GPU-nodes in Figure 11 as a bar chart. The 4 GPUs per node configuration gives about 2.5 times higher performance than running the program only on the CPU cores. And even 1 GPU per node gives significant performance increase compared to the CPU only run.

4 Neurohackathon at Skoltech

The “Zhores” cluster was used as the main computing power during the “Neurohackathon” in the field of neuro-medicine, held in Skoltech from 16-th to 18-th of November 2018 under the umbrella of the National Technology Initiative. It consisted of two tracks: scientific and open. The scientific track included the solution of the tasks of predictive analytics related to the analysis of MRI images of the brain of patients containing changes characteristic of multiple sclerosis (MS). This activity handled private data. Therefore special attention was paid to the IT security. It was necessary to divide the cluster resources such that Skoltech continued its scientific activities while the hackathon participants competed transparently on this facility at the same time.

To address this problem, a two-stage authentication system was chosen using several levels of virtualization. Access to the cluster was made through the VPN tunnel using Cisco ASA and Cisco AnyConnect; then the SSH (RFC 4251) protocol was used to access the consoles of the operating system (OS) of the participants.

The virtualization was provided at the level of a data network through the IEEE 802.1Q (VLAN) protocol and OS level Docker [4] containerization with the ability to con-

nect to GPU accelerators. The container worked in its address space and in a separate VLAN, so we achieved an additional isolation level from the host machine. Also at the Linux kernel level, the namespace feature was turned on and the user and group IDs were remapped to obfuscate the superuser rights on the host machine.

As a result, each participant of the Neurohackathon had a docker container with access via the SSH protocol to the console and used the https protocol to Jupiter application on his VM. The four Nvidia Tesla V100 accelerators on the GPU nodes were used for the computing.

The number of teams participating in the competition had rapidly increased from 6 to 11 one hour before the start of the event. The usage of virtualization technology and the flexible architecture of the cluster allowed us to provide all teams with the necessary resources and start the hackathon on time.

5 Conclusions

In conclusion, we have presented the Petaflops supercomputer “Zhores” installed in Skoltech CDISE that will be actively used for multidisciplinary research in data-driven simulations, machine learning, Big Data and artificial intelligence. Linpack benchmark places this cluster at position 7 of the Russian and CIS TOP-50 Supercomputer list. Initial tests show a good scalability of the modeling applications and prove that the new computing instrument can be used to support advanced research at Skoltech and for all its research and industrial partners.

Acknowledgement: We dedicate this paper to the memory of an outstanding scientist and person, Nobel Prize laureate Zhores Alferov, who passed away on March 1st, 2019. His invention of semiconductor heterostructures led to development of fast interconnect used in all supercomputers. His avid interest for science, his kindness and fairness will without any doubt provide a sparkle for all the users of the supercomputer which was named after him.

Authors acknowledge valuable contribution from Dmitry Sivkov (Intel) and Sergey Kovylov (NVidia) for their help in running Linpack tests during the deployment of the “Zhores” cluster and Dmitry Nikitenko (MSU) for his help in filling the forms for the Russia and CIS Top-50 submission. We are indebted to Dr. Sergey Matveev for valuable consultations and Denis Shageev for indispensable help with the cluster installation. We would also like to thank Prof. Dmitry Dylov, Prof. Andrey Somov, Prof. Dmitry Lakontsev, Seraphim Novichkov and Eugene Bykov

for their active role in organization of the Neurohackathon. We thank Prof. Jacob Biamonte for reading and correcting the manuscript. We are also thankful to Prof. Ivan Oseledets and Prof. Eugene Burnaev and their team members for testing of the cluster.

References

- [1] *Julich mpilinktest*. <http://www.fz-juelich.de/jsc/linktest>. Accessed: 2018-12-15.
- [2] *The BitTorrent Protocol Specification*. <http://www.bittorrent.org>. 2008. Accessed: 2018-12-15.
- [3] *Capabilities of Intel® AVX-512 in Intel® Xeon® Scalable Processors (Skylake)*. <https://colfaxresearch.com/skl-avx512>. 2017. Accessed: 2018-12-15.
- [4] *Docker*. <http://www.docker.com>. 2018. Accessed: 2018-12-15.
- [5] *Environment Modules*. <http://modules.sourceforge.net>. 2018. Accessed: 2018-12-15.
- [6] *Top50 Supercomputers (in Russian)*. <http://top50.supercomputers.ru>. 2018. Accessed: 2018-12-15.
- [7] Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. 2015. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1 19–25.
- [8] BV, ClusterVision 2017. *Luna*. <https://clustervision.com>. Accessed: 2018-12-15.
- [9] Cichocki, A. 2018. Tensor networks for dimensionality reduction, big data and deep learning. In: *Advances in Data Analysis with Computational Intelligence Methods*, 3–49. Springer.
- [10] Hey, A. J. G., Tansley, S., Tolle, K. M., et al. 2009. *The fourth paradigm: data-intensive scientific discovery*, Vol. 1. Microsoft research Redmond, WA.
- [11] Kapral, R., and Ciccotti, G. 2005. Molecular dynamics: an account of its evolution. In: *Theory and Applications of Computational Chemistry*, 425–441. Elsevier.
- [12] Kates-Harbeck, J., Svyatkovskiy, A., and Tang, W. 2019. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature*, 568 7753 526.
- [13] Krapivsky, P. L., Redner, S., and Ben-Naim, E. 2010. *A kinetic view of statistical physics*. Cambridge University Press.
- [14] Lee, H., and Kang, I. S. 1990. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91 110–131.
- [15] Little, J. D. 1961. A proof for the queueing formula: $L = \lambda \cdot W$. *Operations Research*, 9 3.
- [16] Matveev, S. A. 2015. A parallel implementation of a fast method for solving the smoluchowski-type kinetic equations of aggregation and fragmentation processes. *Vychislitel'nye Metody i Programirovanie (in Russian)*, 16 3 360–368.
- [17] Matveev, S. A., Krapivsky, P. L., Smirnov, A. P., Tyrtshnikov, E. E., and Brilliantov, N. V. 2017. Oscillations in aggregation-shattering processes. *Physical review letters*, 119 26 260601.
- [18] Matveev, S. A., Smirnov, A. P., and Tyrtshnikov, E. E. 2015. A fast numerical method for the Cauchy problem for the Smoluchowski equation. *Journal of Computational Physics*, 282 23–32.
- [19] Matveev, S. A., Stadnichuk, V. I., Tyrtshnikov, E. E., Smirnov, A. P., Ampilogova, N. V., and Brilliantov, N. V. 2018a. Anderson acceleration method of finding steady-state particle size distribution for a wide class of aggregation–fragmentation models. *Computer Physics Communications*, 224 154–163.
- [20] Matveev, S. A., Stefonishin, D. A., Smirnov, A. P., Sorokin, A. A., and Tyrtshnikov, E. E. accepted, in press. *Numerical studies of solutions for kinetic equations with many-particle collisions*. In: *Journal of Physics: Conference Series*. IOP Publishing.
- [21] Matveev, S. A., Zagidullin, R. R., Smirnov, A. P., and Tyrtshnikov, E. E. 2018b. Parallel numerical algorithm for solving advection equation for coagulating particles. *Supercomputing Frontiers and Innovations*, 5 2 43–54.
- [22] McCalpin, J. D. 1995. Memory bandwidth and machine balance in current high performance computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, 19–25.
- [23] McVoy, L. W., and Staelin, C. 1996. *Imbench: Portable Tools for Performance Analysis*. In: *Proceedings of the USENIX Annual Technical Conference, San Diego, California, USA, January 22-26, 1996*, 279–294.
- [24] Mei, S., Guan, H., and Wang, Q. 2018. *An Overview on the Convergence of High Performance Computing and Big Data Processing*. In: *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 1046–1051. IEEE.
- [25] Oseledets, I., and Tyrtshnikov, E. 2010. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432 1 70–88.
- [26] Qian, D., and Luan, Z. 2018. High performance computing development in china: A brief review and perspectives. *Computing in Science & Engineering*, 21 1 6–16.
- [27] Seipt, D., Kharin, V., and Rykovanov, S. 2019. Optimizing laser pulses for narrowband inverse compton sources in the high-intensity regime. *arXiv preprint arXiv:1902.10777*.
- [28] Sinitskiy, A. V., and Pande, V. S. 2018. Deep neural network computes electron densities and energies of a large set of organic molecules faster than density functional theory (dft). *arXiv preprint arXiv:1809.02723*.
- [29] Stefonishin, D. A., Matveev, S. A., Smirnov, A. P., and Tyrtshnikov, E. E. 2018. Tensor decompositions for solving the equations of mathematical models of aggregation with multiple collisions of particles. *Vychislitel'nye Metody i Programirovanie (in Russian)*, 19 4 390–404.
- [30] Sukumar, R. 2018. *Keynote: Architectural Challenges Emerging from the Convergence of Big Data, High-Performance Computing and Artificial Intelligence*. In: *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*, 7–7. IEEE.
- [31] Sutmann, G. 2002. *Classical molecular dynamics and parallel computing*. FZJ-ZAM.
- [32] Vallecorsa, S., Carminati, F., Khattak, G., Podareanu, D., Co-dreanu, V., Saletore, V., and Pabst, H. 2018. *Distributed Training of Generative Adversarial Networks for Fast Detector Simulation*. In: *International Conference on High Performance Computing*, 487–503. Springer.
- [33] Van Der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. C. 2005. Gromacs: fast, flexible, and free. *Journal of computational chemistry*, 26 16 1701–1718.
- [34] Zhang, R. 2017. *Applying parallel programming and high performance computing to speed up data mining processing*. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 279–283. IEEE.