Journal of Physics: Conference Series

HPC Resources of the Higher School of Economics

HPC Resources of the Higher School of Economics

P.S. Kostenetskiy, R.A. Chulkevich, V.I. Kozyrev

Higher School of Economics, Moscow, Russia

pkostenetskiy@hse.ru

Abstract. The National Research University Higher School of Economics launched its HPC cluster and created a new division named the Supercomputer Simulation Unit. Now the university HPC cluster occupies seventh place in rating the most powerful computers of the CIS TOP50. The HPC cluster uses to solve machine learning problems, population genomics, hydrodynamics, atomistic and continuous modeling in physics, generative probabilistic models, financial row forecasting algorithms, and other actual problems. Paper describes the HSE HPC resources and experience of their use for scientific and educational tasks.

1. Introduction

In recent years, the world has seen explosive growth in research in the field of artificial intelligence and machine learning. This was greatly influenced by the progress in the development of effective training methods for artificial neural networks, as well as the collection of big data sets. The high level of research carried out at HSE in this direction is confirmed by annual reports at leading international conferences (International conference on machine learning, Conference on neural information processing systems, and Conference on uncertainty in artificial intelligence). The university successfully operates laboratories created in partnership with Samsung (research projects at the intersection of deep learning and Bayesian methods [1]), with PJSC Sberbank (research projects on the application of data analysis in financial technology) and with Yandex (research on the analysis of information from experiments at the Large Hadron Collider [2], research on using privately owned mobile phones as a ground detector array for Ultra High Energy Cosmic Rays [3]). HSE University is a long-term partner of the European Center for Nuclear Research (CERN) and is engaged in improving the efficiency of processing big data with machine learning methods. The university aims to enter the top three national leaders in research and development in the field of artificial intelligence.

To support world-class researches, a modern HPC cluster was installed at the HSE University. Now the university HPC system occupies 7th place in the rating of the most powerful computers of the CIS TOP50 [4]. A new HPC cluster allows the university to carry out high quality researches in deep learning and mathematical modeling. A wide range of methods and applications are being created based of the HPC-complex in the field of high-performance computing, big data analysis, including the intellectual analysis of unstructured and heterogeneous data about the processes of socio-economic and scientific-technological development. To conduct full-fledged research in these areas at the top level, access to a modern software and hardware for processing and storing big data is required.

2. HPC cluster “cHARISMa”

The high-performance computing cluster of the HSE University is called “cHARISMa”, it is an acronym for “Computer of HSE for Artificial Intelligence and Supercomputer Modelling”. The resources of the

HPC cluster are intended to support basic research and teaching at university, as well as to carry out research projects requiring the use of high-performance systems. Cluster resources are allocated upon request for specific projects and a limited period.

2.1. HPC cluster

The HSE HPC cluster consists of 26 specialized computing nodes with large RAM and four modern graphics accelerators Tesla V100 32GB in each node. Technical characteristics of the HSE HPC cluster are presented in Table 1.

Table 1. Technical characteristic of the HSE HPC cluster

Parameter	Value
Number of computation nodes / Processors / Cores / GPUs	26/52/1144/104
Processors model	Intel Xeon Gold 6152 (22x3.70 GHz)
GPUs model	NVIDIA Tesla V100 32 GB
Memory	28 TB DDR4 ECC
Parallel data storage	Lustre, 840 TB
Type of system network	InfiniBand EDR (2x100 Gbit/s)
Type of control network	Gigabit Ethernet
Peak performance	912.4 Teraflops
Performance on the LINPACK	568.5 Teraflops

The presence of four graphics accelerators and a large amount of RAM in each computing node makes it possible to run the tasks of many users on one node simultaneously. Two InfiniBand EDR network cards with a throughput of 100 Gbit/s each are installed in each computing node for data transfer during computing. The InfiniBand network based on the FatTree topology. Also, each node is equipped with a Gigabit Ethernet card for the management network.

The storage system is based on the parallel Lustre file system. The storage system provides up to 15 GB/s writing speed and up to 18.5 GB/s reading speed. At present, the storage system on the HSE HPC cluster consists of five servers: one server management (IML), two metadata servers (MDS), and two storage servers (OSS).

The operating system of the HSE HPC cluster is based on CentOS 7.6 with additional InfiniBand and Lustre drivers. The base system software of the cluster is OpenHPC with Warewulf. It is used to manage boot images for all compute nodes and power control via Redfish.

Several compilers are available. There are GNU GCC/GFortran compilers versions from 7 to 10, and Intel Parallel Studio version 2020 and 2018. To implement MPI features various versions of OpenMPI, MVAPICH and Intel MPI are installed.

The CUDA platform versions 10 and 11 are installed to use graphics accelerators. User environment management and software version switching are implemented using the Lmod package.

Open source, fault-tolerant, and highly scalable cluster management and job scheduling system SLURM version 20 carries out jobs management on the cluster. Its plugins improve the ability to take into account the architectural features of the computing nodes. The GPU scheduling plugin allows to flexibly control the allocation of GPU resources by requesting them in script files along with the CPU and memory. Advanced task management is implemented using the *cgroup* mechanism, which closes most of the possibilities of illegitimate use of resources not allocated to the user.

Various scientific and engineering software is available for users, including Python (Anaconda), GROMACS, GNU Octave, Singularity, etc.

For user management, a user administration system has been developed that is integrated with a university-wide business process management system. This greatly simplifies the process of registering new users, allows tracking ongoing research projects and compiling reports on their implementation.

Widespread Nagios and Ganglia status monitoring systems are installed on the cluster. A custom monitoring system has also been developed that allows to conveniently displaying the current load of the cluster and its schedule.

2.2. Virtualization Cluster

In addition to the high-performance computing cluster, a virtualization cluster was installed to solve virtualization problems with its own software and data storage system. Virtualization nodes are eight nodes with two Intel Xeon Gold 6152 processors (2.1 GHz, 22 cores) and 768 GB of RAM. Two nodes additionally host two NVIDIA Tesla P40 graphics accelerators for forwarding GPUs to virtual machines and supporting virtual desktop infrastructure.

The storage system for the virtualization cluster has three levels of data storage: SSD, HDD with spindle speed 10K, on NS-SAS disks. The total usable storage capacity is 305 TB.

Fiber Channel with two switches is used to connect the virtualization servers to the storage system.

The network core uses two switches with 25GbE and 100GbE ports, combined in MLAG for fault tolerance.

The physical scheme of the computer's network is shown in Figure 1.

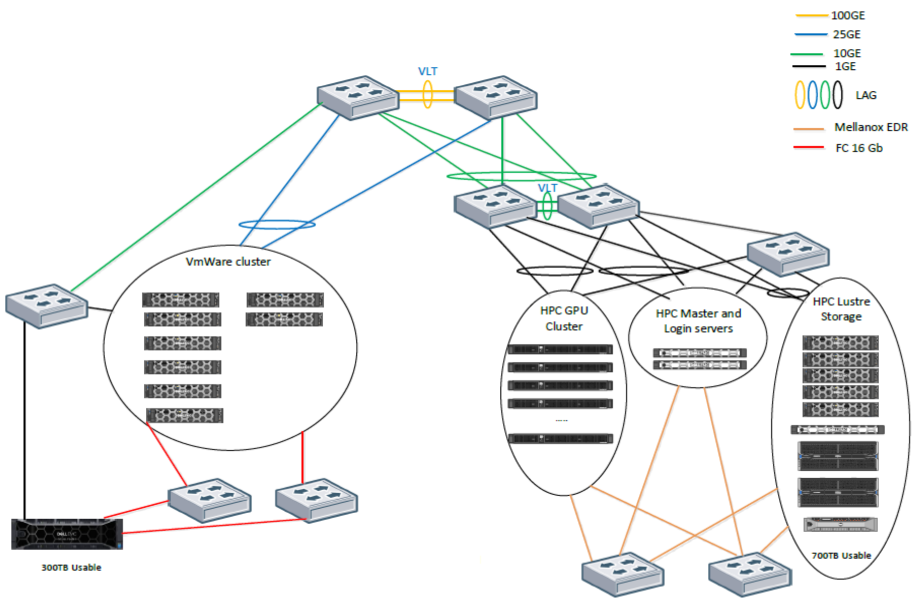


Figure 1. The physical scheme of computer's network

3. Monitoring System

One of the most important tasks of supercomputer centers is monitoring the state of computing resources and the task queue. Proper monitoring allows detecting or predicting possible problems as soon as possible, as well as correcting them, minimizing the impact on users. Today, there are many universal monitoring systems (such as Nagios, Zabbix, Ganglia, etc.), however, almost every HPC cluster is unique and requires something special from the monitoring system.

The Supercomputer Simulating Unit of HSE needs not only to know the state of operability of computing nodes, but also to understand their real workload, see their participation in the queue and the tasks performed on them. Often, HPC clusters use the practice of allocating resources for tasks on a per-node basis, but this is not the way in our case, since our computational nodes have really powerful resources, so many tasks can be performed on one node, each of which uses its own part of these computation resources. This process also needs to be monitored and improved. All these points have encouraged us to implement our own monitoring system that satisfies all of these needs and can make necessary improvements and innovations.

The idea of such a system is simple: processing and logical concatenation of data received from Redfish (IPMI), software counters from the operating system and job queuing system. This allows you to understand the real efficiency of resource use by users, the state of nodes, tasks, and some other things at the same time.

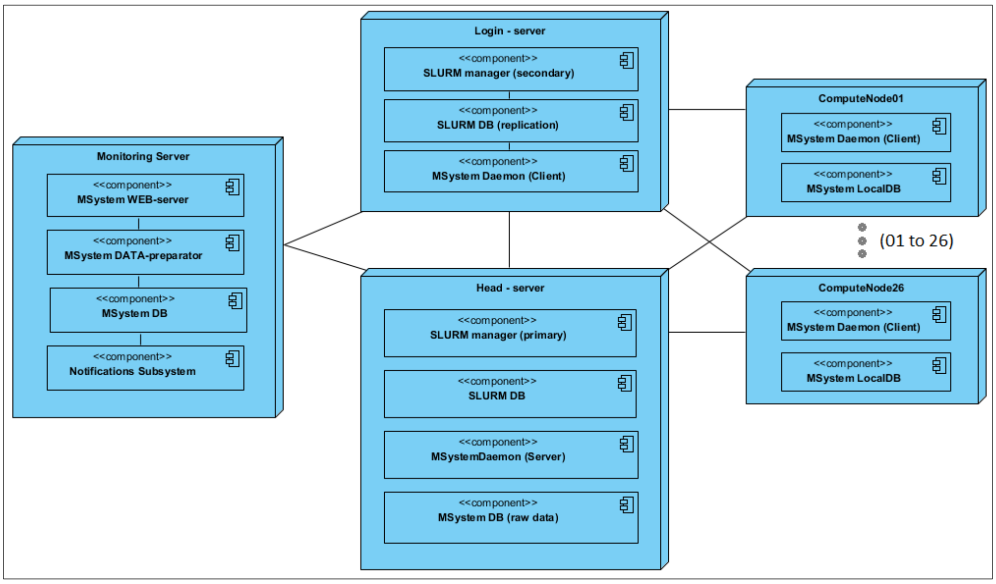
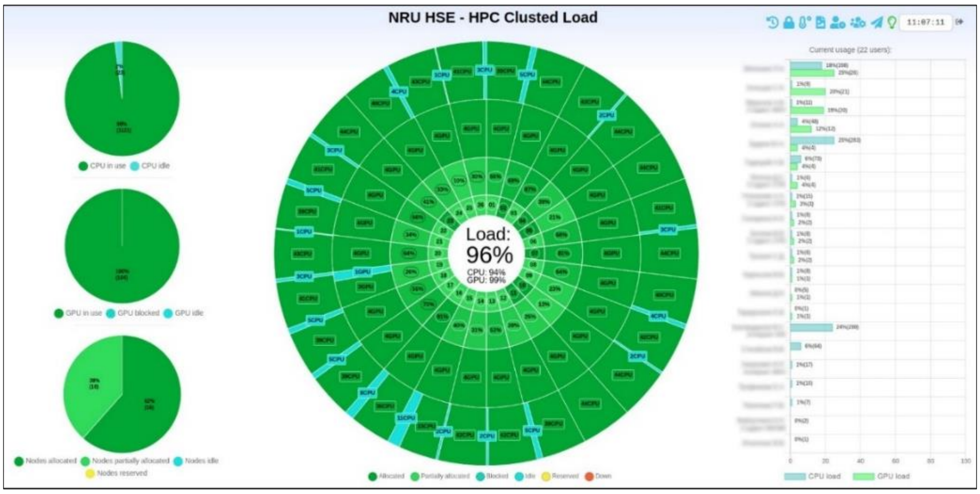


Figure 2. An example of designating the use of a computing node

Figure 2 shows the deployment diagram of the monitoring system components: a daemon launched on each node collects data from meters and IPMI, and transfers it to the head server at some time interval. In the event of any problems that impede the transfer of this data to the head server (for example, network troubles), they are saved locally and synchronized with the host server as soon as this opportunity arises. A daemon executed on the head server saves the information coming from the computing nodes (without significant data-processing). This information, together with data from queue manager, is collected, analyzed, prepared for visual presentation by MSystem daemon and then stored in processed form in the MySQL DB. It is worth noting the fact that replication data about the queue system is also available from the login server. So in the case of late synchronization of data from computing nodes, the system retains the ability to compare and update this data, which, in turn, allows us to evaluate the correctness of user tasks: the ratio of requested and used resources, as well as the effectiveness of this use.

At the moment, the system is still under development, but many things are ready. Figure 3 shows the web interface of the main screen of the system:

1. Three small pie charts (on the left) showing the relationship between the states of the CPU and GPU resources, as well as the state of the nodes in the queue.
2. A horizontal bar chart (right) showing the use of resources by each of the users whose tasks are currently being performed. If you point at the user, the resources used by him will be highlighted on a large pie chart, and when you click on a name, you can get to his personal page.
3. A large pie chart (center) showing the detailed state of the nodes:
   a) internal radius - node numbers;
   b) middle radius - the state of the GPUs on the nodes;
   c) outer radius is the state of the CPU cores at the nodes.



d)

e) **Figure 3.** The main page of the monitoring system

An example of the designation is shown in Figure 4.

The data on the charts are updated in real time (with an acceptable delay of up to 5 seconds). And it is possible to detect the possible optimization queue due to such designations. We have worked with various monitoring systems but have not seen such a presentation of data. As a rule, a large number of graphs makes it possible to understand the state of either the general or an individual calculator but requires several actions. The view developed by us allows for one interaction with the system to determine both the general state of the cluster and detailed information about each computer in it, as well as about the users who use these computing powers and carry out calculations.

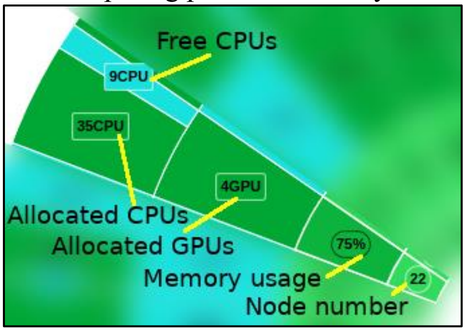


Figure 4. Chart elements

In addition, historical data about cluster loading is often required. The developed system stores all the displayed data, which allows you to visually assess the status of the queues at various points in time, as well as generate loading schedules for reports (example in Figure 5):

Figure 5. Example of an HPC cluster load dynamics diagram

As shown in Figure 5, the system considers three indicators: CPU load, GPU load, and total load. The total load indicator is subjective and, in fact, reflects the loading of available resources in the queue, and not the physical loading of resources. This indicator is calculated by the following formula:

$$L = F + P * (P\text{cpu}+P\text{gpu})/2 \qquad (1)$$

L – total load;
F – percentage of fully occupied nodes;
P – percentage of partially occupied nodes;
P_{cpu} – percentage of occupied CPU cores on partially occupied nodes;
P_{gpu} – percentage of occupied GPU accelerators on partially occupied nodes.

This logic is explained by the fact that (as was already noted above) if all CPU cores are occupied on a computing node (even if all GPUs are free), the computing node is considered to be completely occupied (allocated) from the point of view of the queue.

One of the additional advantages of the system is its interconnection with the internal services of the cluster (user management system, email newsletters, etc.), as well as with external systems, such as the HSE portal. As a result of this integration, the system allows, for example, to generate a report on the use of computing resources by any user or project as soon as it is registered.

The system is implemented using the following technologies: Python, Django, Django REST framework, MySQL, Chart.js (with our improvements) and ApexCharts.js for displaying charts.

4. HPC Cluster Workload

In October 2019, the Higher School of Economics established a new department named *Supercomputer Simulation Unit* for administering the HSE HPC cluster and provide methodological support to users. The extensive experience of the department staff allowed increasing the usability of the cluster, which led to an increase in the number of users and the workload of the cluster (Figure 6).

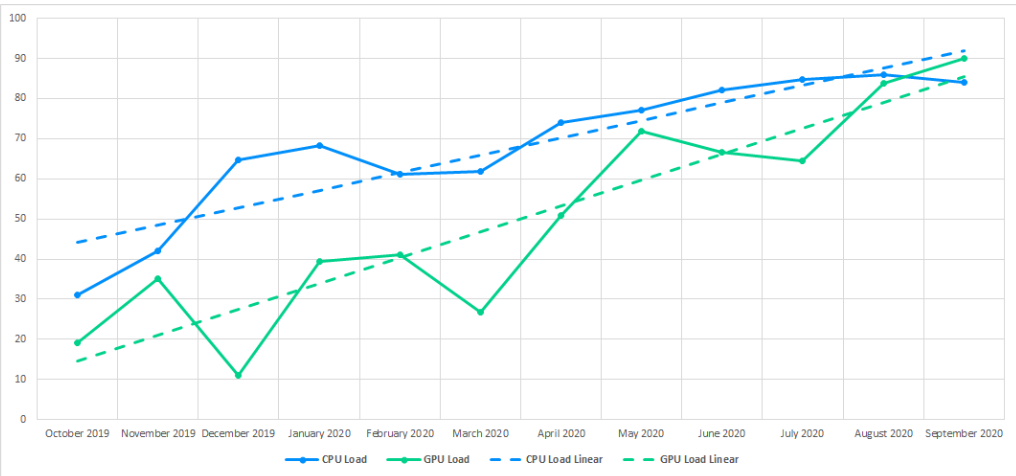


Figure 6. HSE HPC cluster workload growth

The number of HPC cluster users has grown more than 6 times over the year. As of October 2020, the total number of users was 382, of which 195 are students and 187 are HSE employees. The number of units registered on the HPC cluster has grown more than 6 times over the year. As of October 2020, their number was 36. Figure 7 shows the number of scientific and applied projects on the HPC cluster has grown 10 times over the year. Currently, 125 projects are being carried out in various fields of science - medicine, machine learning, chemistry, physics, etc.

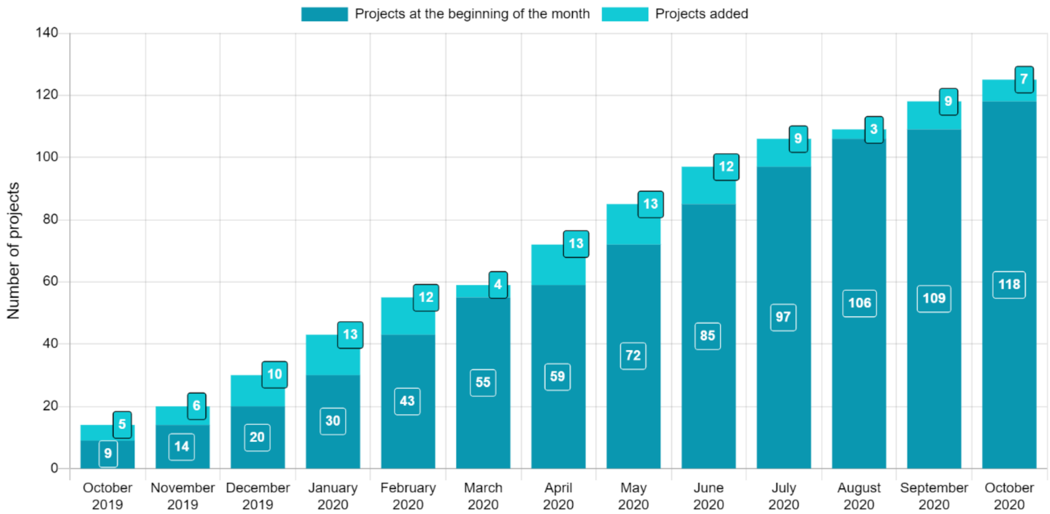


Figure 7. HSE HPC cluster projects number growth

5. Users Tasks

More than 100 important researches are solved on the HPC cluster. Here are listings of some of the projects:

- Carbamazepine solubility in supercritical CO2: A comprehensive study [5];
- Impregnation of Poly(methyl methacrylate) with Carbamazepine in Supercritical Carbon Dioxide: Molecular Dynamics Simulation [19];
- One-Stage One-Shot Object Detection by Matching Anchor Features [6];
- Genomic epidemiology of the early stages of SARS-CoV-2 outbreak in Russia [7];
- Muon identification for LHCb Run 3 [8];
- Transport coefficients of model lubricants up to 400 MPa from molecular dynamics [14];
- Replica redistribution in the implementation of parallel annealing method on the hybrid supercomputer architecture [9];
- Atomistic modeling of condensed matter properties [10],[11];
- Molecular-dynamics study of carbon nanoparticles formation [12],[13];
- Drop oscillation modeling [17];
- Word Sense Induction via LexicalSubstitution for Lexical Semantic Change Detection [18];
- Balanced Identification of the COVID-19 Dynamic Model: General Biological and Country Specific Social Features [20];
- Running Many-Task Applications Across Multiple Resources with Everest Platform [21];
- Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning [22];
- Semi-Conditional Normalizing Flows for Semi-Supervised Learning [23];
- Chaotic Time Series Prediction: Run for the Horizon [24];
- Low-Variance Black-Box Gradient Estimates for the Plackett-Luce Distribution [25].

Supercomputer Simulation Unit employees also do their own research:

- Real-Time System for Automatic Cold Strip Surface Defect Detection [15];
- Simulating of query processing on multiprocessor database systems with modern coprocessors [16].

6. User Administration Software

In addition to various administrative tasks, an important task of supercomputer centers is the registration of new users. This process, usually, includes paper approvals and then user registration in the cluster authorization system and notification of the user about successful registration and transfer of access information to him.

We use an LDAP authorization system. Therefore, when receiving a request for a user's registration, the system administrator must create the user in LDAP and then add to the SLURM database, taking into account all the features of the request (access dates, quotas, scientific project, etc.), and then send a letter to the user. All these tasks may have a human error and therefore must be automated. We have implemented our own registration system that automates all these actions and has reliable checks. It have the interface for the list of new applications to be completed. It is enough for the administrator to click the "Register" button to start the registration process, the status of which is displayed in a pop-up window. After that, a new window appears to send the user an email with registration. The text of the email is generated based on one of the pre-prepared templates and can be also edited before sending.

7. Optimization of computing flow

7.1. Resource limits

One of the most important factors with such a configuration as ours (when up to 44 tasks can be performed simultaneously on one computing node) is the control of the resources involved in the tasks. The fact is that sometimes situations may arise that allow the user to use more computing resources than he was allocated. For example, SLURM Task Scheduler, which is very popular in the HPC industry, has many settings that are not used by default in some systems and settings. The following are examples of the vulnerabilities we encountered.

The first situation arises when the basic definition and restriction of user processes, such as *cgroup*, are not used in the basic SLURM configuration. In this case, after resources are allocated to the user (via *sbatch* or *salloc*) on any computing node, he can go to the node via ssh (in the absence of tasks, the user will be blocked via ssh by PAM) and then use other CPUs and GPU-accelerators by manually starting processes. Also, some software running after the salloc command, even in automatic mode, can take up "extra" CPU cores (for example, *mpirun* from Intel).

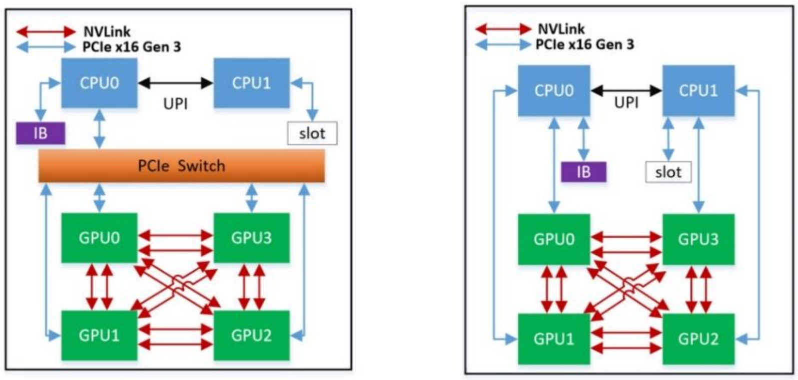
The following situation is that most software has a restriction on the GPU only by the environment variable CUDA_VISIBLE_DEVICES, which the user can redefine in his task and as a result use all available GPUs. However, even if you use the latest version of SLURM (at the time of writing this article is version 20) and have implemented additional protection mechanisms, a task without GPU allocated that does not have the specified environment variable can still access the GPU with index 0 due to the features of CUDA.

Both situations can be easily solved by using the mentioned *cgroup* mechanisms in the configuration, but we still noted them, since *cgroup* was not used on our cluster at the time of installation.

7.2. Server architecture analysis

In addition to universal HPC configurations, it is important to have an idea of such unique (depending on physical server parameters) configurations as the architectural affinity of CPU cores and GPU accelerators, which minimize CPU participation in GPU tasks and maximize performance.

In our case, the computing nodes are Dell C4140, which can have two different architectures, shown in Figure 8. The most performance type is "M", since here, GPU0 and GPU1 are directly connected to CPU0, and GPU2 and GPU3 are connected to CPU1. This is a good case for optimizing tasks with GPUs: all is needed is to take the GPUs that have a direct connection with the CPU allocated for the task [14]. Our servers produced in 2019 are "K"-type architecture. All GPUs are connected to CPU0 through a PCIe switch and do not have a direct connection with CPU1.



(a) Configuration K (b) Configuration M

Figure 8. K and M configurations in DELL C4140 server

Figure 9 shows that only kernels with even indices have architectural proximity with the GPUs, and Figure 10 shows that kernels with even indices are CPU0, which corresponds to the description above.

```
[root@cn-001 ~]# nvidia-smi topo -m
        GPU0    GPU1    GPU2    GPU3    mlx5_0  mlx5_1  CPU Affinity
GPU0     X      NV2     NV2     NV2     NODE    SYS     0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22,24-24,26-26,28-28,30-30,32-32,34-34,36-36,38-38,40-40,42-42
GPU1    NV2      X      NV2     NV2     NODE    SYS     0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22,24-24,26-26,28-28,30-30,32-32,34-34,36-36,38-38,40-40,42-42
GPU2    NV2     NV2      X      NV2     NODE    SYS     0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22,24-24,26-26,28-28,30-30,32-32,34-34,36-36,38-38,40-40,42-42
GPU3    NV2     NV2     NV2      X      NODE    SYS     0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22,24-24,26-26,28-28,30-30,32-32,34-34,36-36,38-38,40-40,42-42
mlx5_0  NODE    NODE    NODE    NODE     X      SYS
mlx5_1  SYS     SYS     SYS     SYS     SYS      X
```

Figure 9. CPU-cores and GPUs affinity

It turns out that ideally we need to run all the GPU tasks only on even cores to achieve the best performance indicators. Now we are just testing this approach. This approach increases the performance of GPU applications by 1-2 percent. Moreover, the benefit is noticeable only when the node is fully loaded with several tasks. By the end of 2020, we plan to upgrade the cluster and buy several “M”-type nodes. Then we will perform comparative testing.

```
[root@cn-001 ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                44
On-line CPU(s) list:   0-43
NUMA node0 CPU(s):     0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42
NUMA node1 CPU(s):     1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43
```

Figure 10. CPU-cores indices

8. Conclusion

Modern scientific research cannot be imagined without the use of high-performance computing facilities and the HSE HPC cluster was installed at the right time. The supercomputer has high performance and a hybrid architecture, which allows the scientific teams of the university to solve various interdisciplinary tasks, as using traditional parallelization approaches, as using deep neural networks and machine learning. The HPC cluster has a large reserve to expansion for further increase its performance. The Supercomputer Simulation Unit was created at the university to administer HPC facilities. The extensive experience of the unit employees allowed to significantly increase the usability of the cluster, the number of active users and the percentage of the total workload. The upgrade of the HSE HPC cluster is planned for 2020 and 2021.

9. Acknowledgments

This research was supported in part through computational resources of HPC facilities at NRU HSE.

References

[1] Molchanov D, Ashukha A and Vetrov D 2017 *Proc. of the 34th Int. Conf. on Machine Learning* (Sydney: PMLR)
[2] Derkach D, Hushchyn M, Likhomanenko T, Rogozhnikov A, Kazeev N, Chekalina V, Neychev R, Kirillov S and Ratnikov F 2018 *J. of Phys.: Conf. Ser.* **1085** 042036
[3] Borisyak M, Usvyatsov M, Mulhearn M, Shimmin C and Ustyuzhanin A 2017 *J. of Phys.: Conf. Ser* **898** 032048
[4] Rating of the most powerful computers in the CIS TOP50, https://top50.supercomputers.ru/list, last accessed 2020/10/30
[5] Kalikin N, Kurskaya M, Ivlev D, Krestyaninov M, Oparin R, Kolesnikov A, Budkov Y, Idrissi A and Kiselev M 2020 *J. Mol. Liq.* **311** 113104
[6] Osokin A, Sumin D and Lomakin V 2020 *Proc. of the European Conf. on Computer Vision* (UK: Springer)
[7] Komissarov A, Safina K, Garushyants S, Fadeev A, Sergeeva M, Ivanova A, Danilenko D, Lioznov D, Shneider O and Shchur V 2020 Genomic epidemiology of the early stages of SARS-CoV-2 outbreak in Russia *Preprint medRxiv 2020.07.14.20150979*
[8] Anderlini L, Archilli F, Cardini A, Cogoni V, Fontana M, Graziani G, Kazeev N, Kuindersma H, Oldeman R and Palutan M 2020 Muon identification for LHCb Run 3 *Preprint* arX*iv:2008.01579*
[9] Russkov A, Chulkevich R and Shchur L 2020 Algorithm for the replica redistribution in the implementation of parallel annealing method on the hybrid supercomputer architecture *Preprint arXiv:20006.00561*
[10] Kondratyuk N, Lenev D and Pisarev V 2020 *J. Chem. Phys.* **152** 191104
[11] Kirova E and Pisarev V 2020 *J. Phys.: Condensed Matter* **33** 034003
[12] Orekhov N, Ostroumova G and Stegailov V 2020 *Carbon* **170** 606–20
[13] Fedorov I, Orekhov N and Stegailov V 2020 *Phys. Rev. B* **101** 100101
[14] Kondratyuk N, Smirnov G, Agarkov A, Osokin A, Nikolskiy V, Semenov A and Stegailov V 2019 *Communications in Computer and Information Science* **1129** 597-609
[15] Kostenetskiy P 2019 *FME Transactions* **47** 765–74
[16] Besedin K, Kostenetskiy P 2014 *Proc. of the Int. Conv. MIPRO* p 1614
[17] Guskova M and Shchur L 2020 *Communications in Computer and Information Science* Drop oscillation modeling
[18] Arefyev N and Zhikov V 2020 *Proc. of the 14th Int. Workshop on Semantic Evaluation* (Barselona: Association for Computational Linguistics)
[19] Gurina D, Budkov Y and Kiselev M 2020 *J. of Phys. Chem. B* **124** 8410–7
[20] Sokolov A and Sokolova L 2020 Balanced Identification of the COVID-19 Dynamic Model: General Biological and Country Specific Social Features *Preprint 10.13140/RG.2.2.31644.92809*
[21] Sukhoroslov O and Voloshinov V 2020 *Communications in Computer and Information Science* Running Many-Task Applications Across Multiple Resources with Everest Platform
[22] Ashukha A, Lyzhov A, Molchanov D and Vetrov D 2020 *Proc. Int. Conf. on Learning Representations* (Ethiopia: OpenReview)
[23] Atanov A, Volokhova A, Ashukha A, Sosnovik I and Vetrov D 2020 Semi-Conditional Normalizing Flows for Semi-Supervised Learning *Preprint arXiv:1905.00505*
[24] Gromov V 2019 *Communications in Computer and Information Science* Chaotic Time Series Prediction: Run for the Horizon
[25] Gadetsky A, Struminsky K, Robinson C, Quadrianto N and Vetrov D 2020 *Proc. of the annual AAAI Conf. on Artificial Intelligence* vol 34 (New York: AAAI Press) pp 10126–35