DATA MINING TO IMPROVE THE EFFICIENCY OF USING THE HYBRILIT HIGH-PERFORMANCE HETEROGENEOUS COMPUTING PLATFORM

E. Polegaeva[1 a], D. Priakhina[1,2], O. Streltsova[1,2], D. Podgainy[2]

[1] *Dubna State University, Russia, Moscow region, Dubna, 141980, 19 Universitetskaya*

[2] *Joint Institute for Nuclear Research, Russia, Moscow region, Dubna, 141980, 6 Joliot-Curie*

E-mail: [a] robin_goul@mail.ru

The HybriLIT heterogeneous computing platform is part of the Multifunctional Information and Computing Complex of the Meshcheryakov Laboratory of Information Technologies of the Joint Institute for Nuclear Research. An analysis of data on the use of the HybriLIT platform is carried out: special attention is paid to the study of information about the resources used when starting tasks by various users and the time of their implementation. The relevance of this study lies in the ability to predict the further workload of the platform based on the analysis obtained, which will enable the more rational and efficient use of not only the available computing resources, but also the resources of data storage systems. This paper presents models for predicting the usage of the HybriLIT resources based on data through analysis. Several machine learning methods are compared to choose a model that gives the best prediction accuracy.

Keywords: heterogeneous computing platform, data analysis, machine learning, data prediction

Ekaterina Polegaeva, Daria Priakhina, Oksana Streltsova, Dmitry Podgainy

1. Introduction

The HybriLIT heterogeneous platform is part of the Multifunctional Information and Computing Complex, the Meshcheryakov Laboratory of Information Technologies of JINR, Dubna [1]. The platform consists of the "Govorun" supercomputer and the HybriLIT training and testing polygon, which have a single software and information environment. The HybriLIT platform is designed for resource-intensive and massively parallel calculations and has both central processors and graphics accelerators of different types.

Various services are being developed on the HybriLIT platform, they allow monitoring the load of system components (compute nodes, storage systems, virtual machines, etc.), and obtaining aggregated data for analyzing the efficiency of resource use. An important addition to the development of services would be an analytical system that will make it possible to predict the use of various types of resources and enable deeper data analysis. This paper presents a study, a direction for the development of algorithms for such a system using machine learning methods to analyze data on user tasks both for the resources used and for various user groups, as well as to solve the problem of predicting the requested resources and their types. The study was carried out on the data of user tasks of the HybriLIT training and testing polygon, full information about which is collected by the SLURM scheduler and resource manager [2] installed on the platform.

The database contains data from April 2018 to August 2020 (29 months), based on which deep data analysis was performed and machine learning models were trained and tested. The work was carried out in the Python language in the Jupyter notebook development environment deployed on the HybriLIT resources. Modules used were pymysql, numpy, pandas, matplotlib, sklearn.

2. Data Analysis

Data analysis and visualization are necessary for a more convenient perception by users of information about the platform load.

It is noteworthy that the data collected in the SLURM database for the period under review contains thousands of rows, information about the CPU cores used and the number of GPU accelerators involved is collected for each task. At the same time, on the HybriLIT testing polygon, computational nodes contain different types of CPU and GPU. For this study, performance differences between different CPUs and GPUs are not considered.

Data analysis was carried out earlier, its results are described in [3]. Analysis and visualization were prepared for the following tasks:

- Number of nodes used on various resources;
- Number of logical cores used on different nodes;
- Number of running tasks on different nodes;
- Duration of tasks running on nodes;
- Duration of tasks launched by each user group;
- Number of logical cores used by each user group;
- Total number of resources used on the platform;
- GPU resources usage;
- Number of logical cores used when performing tasks on GPU resources;
- Number of compute nodes and logical cores used over time;
- Number of logical cores used on CPU and GPU resources for the entire period of time;
- Number of running tasks for the entire period of time;
- Duration of launched tasks for the entire period of time;
- Use of GPU resources for the entire period of time;

These tasks for analysis were selected to see the workload of computational nodes, to identify the most and least active user groups, and track changes in the resource load over time. The latter criterion will be used in the future as the basis for predicting the further workload of the platform.

Further predictions will be based on the results of the analysis of the problem concerning the number of CPU cores and GPU resources used by all users for all the time [fig. 1].

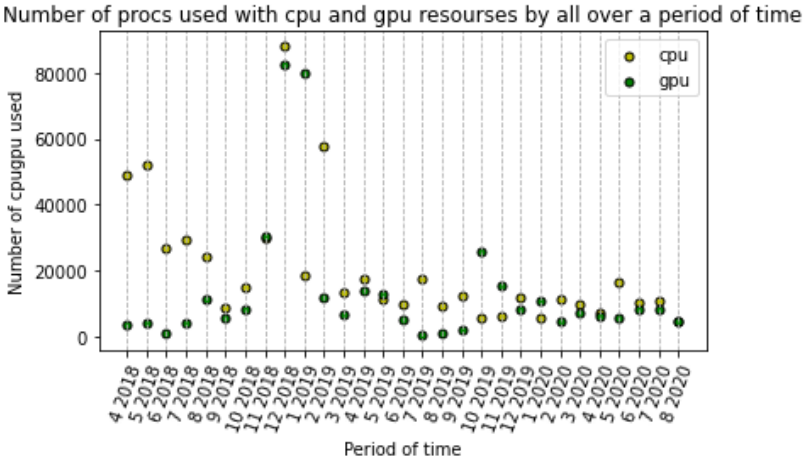

Figure 1. Analysis of the amount of CPU and GPU resources used

3. Prediction and Verification

Predicting future resource use is a regression task. To obtain correct values, it is necessary to use relatively homogeneous data, as peak jumps introduce large errors. Since the identification of anomalies is not included in the presented task, such data is not taken into account. The maximum peak in resource utilization occurred in December 2018 and subsequent months. Therefore, for making predictions, data is not used for the entire period.

To build predictions and verify them, the data were split into training and test data. The training data was taken from April 2018 to March 2020. The predictions were based on the period from April 2020 to August 2020. Metrics such as mean absolute error (MAE), mean square error (MSE) and mean absolute percentage error (MAPE) were used to evaluate the accuracy of the predictions. The accuracy was estimated both for the first month of predictions and for the entire period. Machine learning models such as linear regression, polynomial regression, classification and regression trees, random forest, and XGBoost were chosen to build the predictions.

Since the task under study is regression, the models of classical supervised learning were first considered. The first machine learning model used was linear regression, which is described in more detail in [4]. Despite the fact that it is obviously not suitable for this task, its use was necessary to confirm the nonlinearity of the data. The results of the model showed that it could not be used in this task [fig. 2]. Next, we used polynomial regression, which handles non-linear data much better [5]. However, it did not give a satisfactory result either [fig. 3].

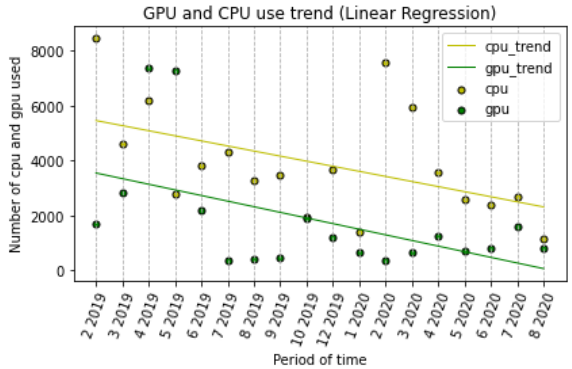

Figure 2. Linear Regression Trendlines

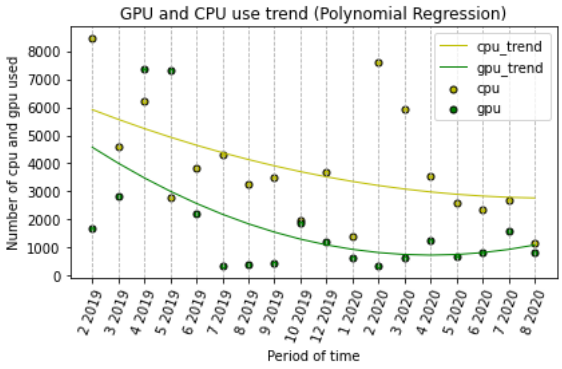

Figure 3. Polynomial Regression Trendlines

Then a model of classification and regression trees [6] was built. With an excellent result on the training set, the predictions did not give the best result [fig. 4].

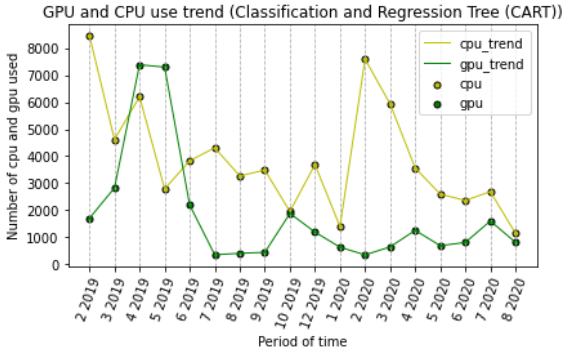

Figure 4. Classification and Regression Trees Trendlines

Proceeding from the fact that the models of classical learning did not give a satisfactory result and showed an accuracy of less than 90%, ensemble methods were applied. The random forest model, which refers to bagging models [7], was used. It gave a more than satisfactory result and showed an accuracy of 94%. The peculiarity of the random forest model is that it is based on decision trees, but randomly selects features for further division and does not go too far, excluding the possibility of overfitting the model [fig. 5]. In search of the best result, the XGBoost boosting model [8] was applied. As expected, its results exceeded the accuracy of the random forest and gave predictions with an accuracy of 99%, which makes this model the most suitable for solving the task [fig. 6].

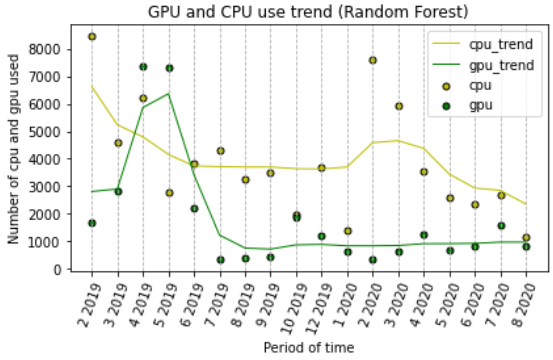

Figure 5. Random Forest Trendlines

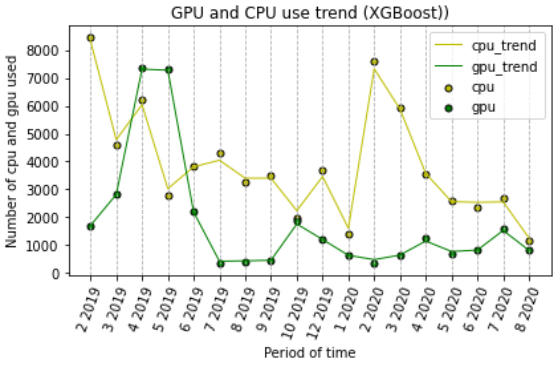

Figure 6. XGBoost Trendlines

For each model built, data was collected to assess the accuracy of the predictions. A comparison of the predicted and actual values was carried out, the accuracy of predictions for short and long-term periods was estimated, and the average prediction accuracy for April 2020 was given [tab. 1].

Table 1. Results and prediction accuracy

Machine learning model	**Resource**	**April 2020**		**Accuracy**		**Average accuracy April 2020**
		Prediction	**Real**	**April 2020**	**Apr. 2020 – Aug. 2020**	
Linear Regression	CPU	3708	3563	96%	27%	53%
	GPU	2375	1255	11%	0%	
Polynomial Regression	CPU	2956	3563	83%	76%	47%
	GPU	2375	1255	11%	0%	
Classification and Regression Trees	CPU	3691	3563	96%	28%	87%
	GPU	980	1255	78%	71%	
Random Forest	CPU	3318	3563	93%	42%	94%
	GPU	1323	1255	95%	51%	
XGBoost	CPU	3563	3563	100%	34%	99%
	GPU	1246	1255	99%	58%	

4. Conclusion

Deep data analysis was carried out, it enables to assess the current resource load. A study was also conducted to select a machine learning model that best suits the task at hand. As a result, an algorithm was obtained, it makes it possible to get predicted values on the use of CPU and GPU resources for the next month with an accuracy of 99%. Thus, the presented work is of great practical importance. The resulting predictions will allow system administrators to make decisions on load redistribution, as well as on the purchase of new equipment, if required.

The analysis and prediction of the platform load is a relevant topic and has recently been taken into development, therefore, the field of activity on this issue is wide. Plans for further work include, for example, improving prediction algorithms in order to enhance the accuracy of long-term predictions. It is planned to predict the use of platform resources not only in general, but also separately for each user group. In addition, in the future, users will be clustered to identify their non-obvious groups. The obtained algorithms will be applied on other computing platforms of JINR, in particular, on the "Govorun" supercomputer.

References

[1] HybriLIT heterogeneous platform. Available at: hlit.jinr.ru

[2] SLURM workload manager. Available at: https://slurm.schedmd.com/documentation.html

[3] Polegaeva E., Priakhina D., Streltsova O. Analysis of data on the loading of high-performance platforms by user tasks on the example of the heterogeneous computing platform HybriLIT, 2021;(2):67-76 (In Russian). Available from: http://sanse.ru/download/437

[4] Demidenko E.Z. Lineinaya i nelineinaya regressii // M.: Finansy I statistika. 1981. P.302

[5] Pant A. Introduction to Linear Regression and Polynomial Regression [towards data science]. Available at: https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb (accessed 05.05.2021)

[6] Brownlee J. Classification And Regression Trees for Machine Learning [Machine Learning Mastery]. Available at: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/ (accessed 23.10.2020)

[7] Donges N. A complete guide to the random forest algorithm [Data science]. Available at: https://builtin.com/data-science/random-forest-algorithm (accessed 23.10.2020)

[8] Brownlee J. XGBoost for Regression [Machine Learning Mastery] Available at: https://machinelearningmastery.com/xgboost-for-regression/ (accessed 20.06.2021)