Long-term prediction of cloud resource usage in high-performance computing

Piotr Nawrocki[1][0000−0003−4512−9337] and Mateusz Smendowski[1]

AGH University of Krakow, Poland
piotr.nawrocki@agh.edu.pl, smendowski@student.agh.edu.pl

Abstract. Cloud computing is gaining popularity in the context of high-performance computing applications. Among other things, the use of cloud resources allows advanced simulations to be carried out in circumstances where local computing resources are limited. At the same time, the use of cloud computing may increase costs. This article presents an original approach which uses anomaly detection and machine learning for predicting cloud resource usage in the long term, making it possible to optimize resource usage (through an appropriate resource reservation plan) and reduce its cost. The solution developed uses the XGBoost model for long-term prediction of cloud resource consumption, which is especially important when these resources are used for advanced long-term simulations. Experiments conducted using real-life data from a production system demonstrate that the use of the XGBoost model developed for prediction allowed the quality of predictions to be improved (by 16%) compared to statistical methods. Moreover, techniques using the XGBoost model were able to predict chaotic changes in resource consumption as opposed to statistical methods.

Keywords: Cloud computing · Resource prediction · High-performance computing · Machine learning.

1 Introduction

Cloud computing is increasingly becoming an alternative to supercomputers for some high-performance computing (HPC) applications [6, 8]. Among the potential use areas of cloud computing are various advanced simulations such as, for instance, HPC simulations of disease spread or social simulations [21]. Hybrid environments are also being used, where part of the sensitive computation is performed locally and part in a public cloud [14]. Public clouds can be used where the demand for computing resources is greater and cannot be satisfied by local resources. Of course, the cost of using cloud resources in HPC has to be taken into account, which is why the issue of optimizing their use is so important. In this context, it is very important to be able to predict the consumption of cloud resources in order to reserve them optimally. Reserving too many resources may result in increased costs and reserving too few resources may cause problems with simulation execution. Although cloud resource usage prediction (and

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

subsequent scheduling) is discussed in the literature, it is almost always in the context of autoscaling and short-term prediction. For HPC applications such as advanced simulations, many tasks may require resources in the long term, and therefore the time horizon of mechanisms such as autoscaling and short-term prediction may prove too short for this type of computing. The long-term prediction solution developed by the authors (using machine learning) allows for a longer prediction horizon, which can be useful for advanced simulations using HPC.

The major contributions of this paper can be summarized as follows:

- designing a solution that provides long-term resource usage predictions for advanced simulations using HPC;
- developing a self-adapting system that can operate in production-grade environments with long-term load changes;
- conducting evaluation using data collected from a real-life production system;
- comparing the results obtained using different prediction techniques based on the XGBoost model with statistical methods.

The rest of this paper is structured as follows: Section 2 contains a description of related work, Section 3 focuses on the description of a long-term cloud resource usage prediction system, Section 4 describes the experiments performed, and Section 5 contains the conclusion and further work.

2 Related Work

There are many publications dealing with the use of cloud computing resources for HPC applications. In [7], the author discusses the use of a virtual cluster for this type of computing (using Elastic Computing Cloud – EC2 as an example). Clusters of computer systems are a common HPC architecture. However, small local clusters, although cost-effective, may not be efficient enough. On the other hand, large clusters are more expensive and it is not always possible to provide them with a sufficient sustainable load. Therefore, virtual clusters using cloud resources offer a viable alternative. In this case, public commercial cloud environments provide users with storage and CPU power to build their own dedicated clusters of computers that can be used in scientific computing applications. Based on the experiments performed, the author shows that it is possible to use a virtual cluster to realize various computations including HPC. The author also analyzes the cost of using cloud solutions; however, he does not explore the possibility of optimization of cloud resource consumption.

In [5], the authors discuss scalability, interoperability, and achieving guaranteed Quality of Service (QoS) in a High-Performance Computing Cloud (HPCC) environment. The authors propose a cloud resource management framework to handle a large number of user HPC application requests and manage multiple cloud resources. System tests were performed using a large number of real-world

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

HPC applications. To evaluate the performance of the system proposed, the authors used performance metrics such as response time, the number of requests successfully handled and user satisfaction. What is missing from this work, however, is an analysis of the cost of using cloud resources using the system developed and the possibility of optimizing them.

In [17], the authors present the possibilities of using HPC in the Google Cloud Platform in emergency situations when efficient processing of large amounts of traffic data is required. This allows for effective disaster management using massive data processing and high performance computing. Another application of HPC in cloud computing is presented in [9] where the authors describe a platform for computer vision applications that enables audio/video processing and can use high performance computing cloud resources for this purpose.

An extensive analysis of the possibilities of cloud solutions related to High Performance Computing, among other things, is presented in [2]. The authors analyze the capabilities of the four most popular environments – Amazon Elastic Compute Cloud, Microsoft Azure Cloud, Google Cloud and Oracle Cloud. Today, HPC workloads are increasingly being migrated to the cloud. This is due, among other things, to the flexible nature of the cloud where resources can be expanded and reduced on demand, which optimizes the cost of using cloud computing. At the same time, computationally intensive workloads can be performed efficiently on cloud resources that are connected via a high-speed network. The most suitable cloud model for HPC users is Infrastructure as a Service (IaaS), where it is possible to specify all the details of the infrastructure needed to create clusters. There are many areas where the cloud is being used for HPC, including financial risk simulations, molecular dynamics, weather prediction, and scientific or engineering simulations.

As it can be gathered from the above analysis, the use of cloud resources for HPC is now commonplace. There are also appropriate mechanisms for managing cloud resources so as to adjust them for HPC purposes and, for instance, predict the placement of containers [1]. However, there is a lack of mechanisms to optimally reserve cloud resources for HPC, especially in the long term. One way to conduct such optimization is to use prediction mechanisms to determine what resources will be required in the future for HPC and establish an optimal reservation plan. The most common strategy for using cloud computing resources is to reserve resources at the highest potential demand level, which, however, generates unnecessary costs for unused cloud resources. Predictive resource utilization can prevent this, helping to reserve only the cloud resources needed. However, the majority of research on cloud resource management and optimization concerns autoscaling [20, 4], predictive autoscaling (autoscaling with workload forecasting) [18, 22] and short-term prediction [19, 3]; there are very few studies of long-term prediction.

The authors' earlier works indicate that cloud resource usage prediction makes it possible to create optimal plans for resource reservation, leading to significant savings. Those works included research on using machine learning and adaptive resource planning for cloud-based services [10], anomaly detection

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

in the context of such planning [13], cloud resource usage prediction mechanisms taking into account QoS parameters [11, 15], data-driven adaptive cloud resource usage prediction [12] and resource usage cost optimization in cloud computing [16]. However, none of the previous works analyzed HPC needs in the context of cloud computing resources. In the research described in this article, the authors paid particular attention to the need for long-term prediction (relevant to HPC) and used a full data set from more than one year to develop the model. In previous work (such as in [16, 15]), only selected data from three months were considered. In addition, the XGBoost model was used, in which prediction was conducted for one-week periods with feature enrichment: using *lagged features* from the previous week, features derived from the Fast Fourier Transform (FFT), moving window statistics and calendar features, taking into account holidays.

3 Long-term cloud resource usage prediction system

The prediction system developed was designed to forecast weekly CPU usage by predicting 168 sample resource usage metrics with hourly resolution. The system consists of seven modules and the starting point for its operation is the use of historical data to obtain an initial prediction (Figure 1).

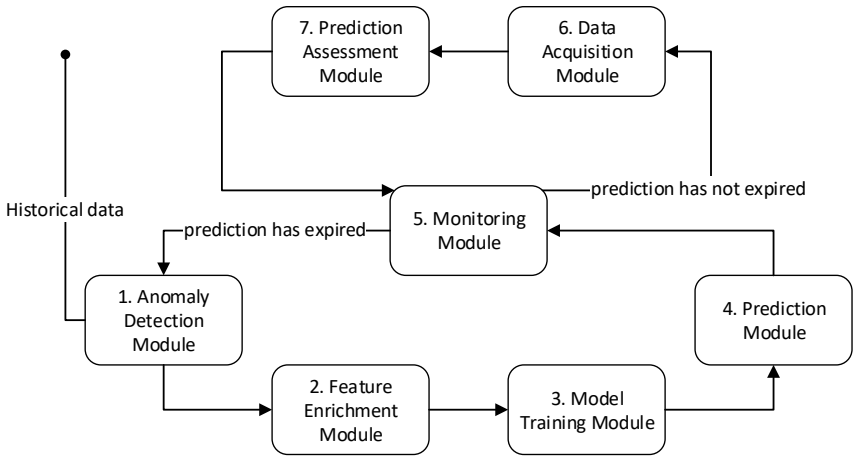

Fig. 1. Prediction system concept

At the start of prediction system operation, historical values of CPU usage metrics are recorded and fed into the *Anomaly Detection Module*. This module is flexible in terms of the ranges of data to which the anomaly detection model is applied. It is noteworthy that anomalies are understood as samples that are outlying in the data, effectively translating into drastic resource consumption peaks.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

Conceptually, anomaly detection can be applied either to the entire dataset or locally to smaller subsets of data independently. Such a configuration enables conducting multiple simulations and checking whether the anomalous data detected are undesirable or constitute a valuable contribution. Subsequently, the *Feature Enrichment Module* is responsible for enriching data in diverse ways to create an expressive data representation. For time-series, there are several techniques that can be implemented at that level to extract maximum valuable information from the dataset, which will positively impact the results of the prediction model. The training of the prediction model itself is carried out through the *Model Training Module*. After completing the training of the model and its initial evaluation on the validation set, either through walk-forward validation technique or on a separate validation set, long-term prediction for a weekly period is performed. Multi-step prediction can be achieved in various ways, with direct, iterative, and multi-output prediction being among the most popular. After performing the prediction using the Prediction Module, the system transitions from the outer loop to the inner loop. The *Monitoring Module* checks whether the prediction is valid at hourly intervals, which correspond to the granularity intervals of the recorded metrics. If the prediction has not expired and is valid, the resource usage level is recorded and metrics are stored (*Data Acquisition Module*) for the indicated time period, and the prediction previously made is verified against actual usage (*Prediction Assessment Module*). When the *Monitoring Module* detects that the expiration time has been reached, the system exits the inner loop and returns to the outer loop of the prediction system. At this point, the system has acquired new data, which can be scanned for anomalies, enriched and used to re-adapt the prediction model. This is especially important for dynamic, chaotic time series where the model needs to adjust to the current data distribution. In this way, the system can operate in a continuous optimization loop, adapting to newly recorded data and making predictions for the configured horizon. The specific implementation employs Isolation Forest and XGBoost as the main models in the anomaly detection and prediction modules, respectively. However, the overall concept is generic and flexible, allowing for the use of various techniques within the individual modules. Flexibility and self-adaptation are especially important in the context of HPC resources, where different prediction schemes can be evaluated to select the appropriate one regarding future resource reservations for supercomputing.

4 Evaluation

As part of the evaluation of the prediction system, historical records of CPU usage were utilized, with a particular focus on this metric among the various options available from cloud monitoring services. This data is considered crucial in the realm of high-performance computing, alongside Random Access Memory (RAM) usage. The data were collected from a real-life production environment. The recorded CPU usage metrics are closely tied to a timestamp, thus the data are represented in the form of a time series. Ultimately, historical records rep-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

resent consolidated CPU utilization over a period of exactly 80 weeks, where metric values were registered at 1-hour intervals. Moreover, several sequences in the acquired dataset were found to have missing data, which were imputed using a Simple Moving Average (SMA) approach with a window size equal to the prediction horizon of 168 samples. The starting point for research was analyzing the data, specifically by arbitrarily partitioning them into training, validation, and testing sets. This was done to enable a focused application of methods to the training set, performance evaluation using the validation set, and critical assessment of generalization abilities without unwanted knowledge leakage from the testing set during the training process. Due to high variance in the dataset and significant changes in the characteristic features that amplify the chaotic nature of the time series, effectively distinguishing the sizes of the datasets proved to be a challenge. Ultimately, 10,920 and 2,520 samples were used for the training and testing sets, respectively. Additionally, the last 840 samples from the training set were selected as the validation set. In order to examine the nature of the dataset, a Jarque-Bera test was conducted on the training set (without the validation portion) to assess its normality and skewness. The initial output value (58.96), represented the test statistic of the Jarque-Bera test. This was used to evaluate the deviation of the sample data from a normal distribution, with a higher test statistic indicating a greater deviation from normality. The second value, 1.58×10^{-13}, represents the p-value of the test, which suggests strong evidence against the null hypothesis that the data is normally distributed. Furthermore, the stationary test indicated non-stationarity, thereby implying that the variable under investigation does not exhibit constant statistical properties over time.

The main prediction model utilized in the system developed is XGBoost, which was optimized for evaluation using grid search with selected parameters of 240 estimators, 7 as the maximum depth, and a learning rate of 0.28. To enhance prediction accuracy, various features were incorporated, including lagged features from the preceding week, exogenous features derived from the calendar, such as day of week, month, hour, holidays, and statistical parameters calculated within the scope of a moving window. Two sizes of moving windows (168 and 72) were selected to capture statistics from the last week and the last three days, respectively. The statistical features included classical metrics such as the median and the mean as well as more sophisticated statistics such as interquartile differences, values above the mean, skewness, kurtosis, and features resulting from decomposing the time series into the frequency domain using Fast Fourier Transform. Additionally, cyclical features from all calendar parameters were enriched using the cosine and sine functions. Cumulatively, the feature enrichment phase made it possible to obtain 250 features. The XGBoost algorithm was trained for single-step-ahead prediction. To enable long-term forecasting, an iterative approach was employed where each prediction was used as a training sample in the following step until the end of the prediction horizon. Features were computed for each sample in real time, in a manner preventing information leakage.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

In the context of long-term prediction, identifying outlying samples that are understood as drastic and local fluctuations in the resource usage that do not translate into a permanent change in characteristics, and determining whether they contribute to the model as anomalies or provide valuable information are crucial factors. In order to tackle this challenge, we employed an unsupervised Isolation Forest model for evaluation. The parameters of the Isolation Forest model were also determined using grid search, with a selection of 90 estimators and a contamination fraction of 0.05. Other parameters were left at default values to reduce the search space, which, in the case of walk-forward validation, was less computationally complex. The strategy selected for handling outlying samples was their removal. Imputing the mean value of a certain horizon of samples was considered as well, but was not utilized due to the rich feature engineering approach that also took into account statistical parameters calculated within the windows in question.

Prior to evaluating the implemented prediction system based on the XGBoost model, certain reference results were established using statistical methods, which served as a baseline. Typically, naive prediction (repeating the last observed value throughout the entire prediction horizon) or more sophisticated methods such as Simple Exponential Smoothing serve as a solid reference point. However, for comparison purposes, neither of the aforementioned methods nor other methods such as Holt's Exponential Smoothing with a damped trend were used. Although these methods can produce good error metric values, they do not reflect the data characteristics that are crucial for resource demand prediction in HPC. Consequently, Holt-Winter's Seasonal Smoothing methods were used as baselines, namely *Method 1* (with a 1-day seasonality), *Method 2* (with a 3-day seasonality), and finally *Method 3* (with a 1-week seasonality). At the end of each prediction horizon, statistical methods were re-adapted to account for the new data. The listed seasonality periods to be explored were determined by analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) as well as by decomposing the data into trend and seasonality, both in terms of additive and multiplicative models. However, at the data exploration stage, it was discovered that no singular principal seasonal/frequency component was present, a finding which was additionally confirmed using the FFT (Fast Fourier Transform) and DWT (Discrete Wavelet Transform) techniques.

Figure 2 presents the forecasting results using three baseline methods. In cases where the time series exhibit stability over a given period, simple methods can adequately reflect utilization values. However, naive methods and the usual statistical models that often assume linear and stationary data, while performing well when the characteristics of the data are stable, may fall short in predicting sudden fluctuations, chaotic increases or decreases in resource utilization, which can be observed in Figure 3. As far as the analysis of error metrics is concerned, it should be carried out in tandem with the examination of trend characteristics to objectively assess the generalization capabilities of methods as well as their strengths and potential weaknesses where inefficiencies may arise. Table 1 summarizes the error metric values obtained for the entire testing set

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

for baseline methods. *Method 2* has the lowest RMSE (Root Mean Squared Error) and NRMSE (Normalized RMSE) values, which indicate a lower overall error in prediction compared to the other methods. Additionally, *Method 2* has a slightly higher MAE (Mean Absolute Error) compared to *Method 1*. On the other hand, *Method 3* exhibits the poorest performance, except for the MAPE (Mean Absolute Percentage Error), where typically the highest amplitudes and oscillations in forecasts can be observed, and in majority of cases forecasts are below ground truth. The MdAE (Median Absolute Error) indicated a clear advantage for *Method 1*. Finally, based on this analysis, *Method 1* and *Method 2* yield comparably promising results. However, despite their relatively good prediction results in terms of error metrics, these methods completely failed to handle sudden changes in resource consumption. This provides a strong motivation to explore more advanced prediction systems or models, particularly in HPC systems where overprovisioning leads to high costs and underprovisioning results in undesired drops in QoS. The usage of such baselines in production may heighten the risk of user service unavailability.

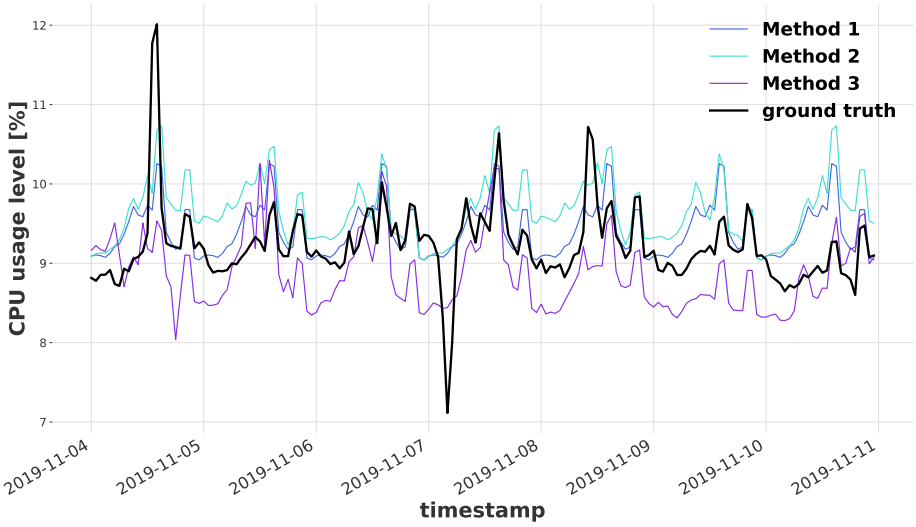

Fig. 2. CPU usage prediction results for baseline methods (first sample period)

Table 1. Error metrics for baseline methods obtained for the entire test set

Reference	RMSE	NRMSE	MAE	MAPE	MdAE
Method 1	2.223	0.141	0.914	0.267	0.327
Method 2	2.168	0.137	0.936	0.276	0.374
Method 3	2.339	0.148	0.999	0.261	0.415

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

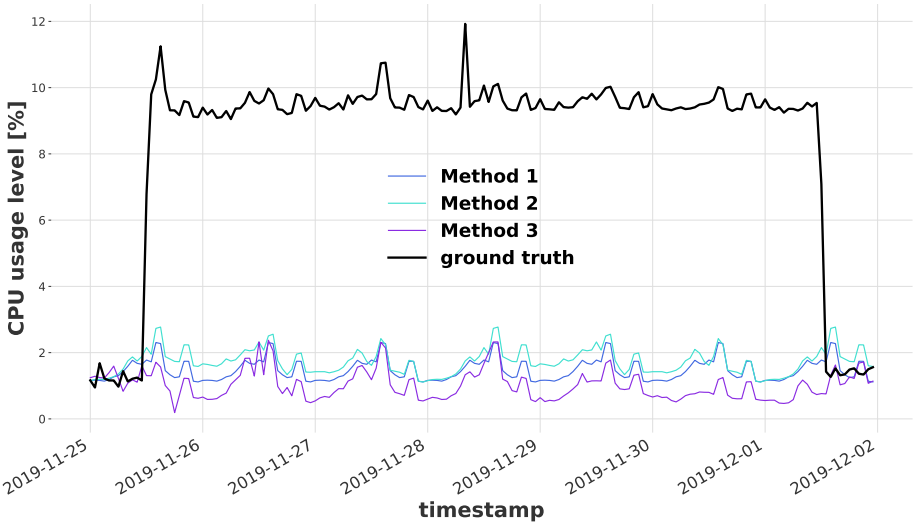

Fig. 3. CPU usage prediction results for baseline methods (second sample period)

Following the conclusions drawn from baselines and their limited applicability, in order to evaluate the prediction system based on the XGBoost model, multiple different simulations and comparisons were conducted, resulting in eleven prediction methods that serve as a robust foundation for drawing conclusions. When evaluating diverse prediction methods, specific components of the prediction system were selectively utilized, as some of the tested prediction methods did not employ the anomaly detection module. Firstly, *Method 4* represents the basic approach, which utilizes a one-week prediction system without the anomaly detection module and without XGBoost model retraining (thus without adapting to newly acquired CPU usage metrics). Subsequently, *Method 5*, *Method 6* and *Method 7* extend the previous approach with anomaly detection within the training set and handling such anomalies using the removal strategy. The differences between these methods concern the scope of anomaly detection: the detection was applied to the entire training set, 4-week portions and weekly portions, respectively. In the context of time series data, it is crucial to determine whether samples identified as anomalies introduce noise and perturbations to the prediction models or rather provide valuable information for generalization purposes. Furthermore, *Method 8* involves refitting the model after the weekly prediction validity period expires, while taking into account newly collected metrics. However, the anomaly detection module was not involved. Subsequently, *Method 9*, *Method 10* and *Method 11* extend the periodical (weekly) re-training approach but include anomaly detection also for the entire training set, 4-week and 1-week portions, respectively. Finally, *Method 12*, *Method 13* and *Method 14* are basically extensions of *Methods 9*, *10* and *11*, but Anomaly Detection is additionally applied to newly collected data before each refit so all data on which the model

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

is trained are scanned against the outliers. Figure 4 indicates that in the case of stable resource usage values, it is evident that the prediction characteristics of the methods applied closely match real-life CPU consumption values from the test set. Particularly noteworthy and inspiring results that lie at the heart of the problem are presented in Figure 5, where the methods predicted an increase in consumption, approaching the actual recorded consumption, which was entirely missed by the baseline. The predictions dynamically followed the consumption trend, with various methods reacting differently but each responding to the possible change in resource usage trend. This indicates a huge advantage of the implemented prediction model over baselines, which results from its dynamics and reactive attitude.

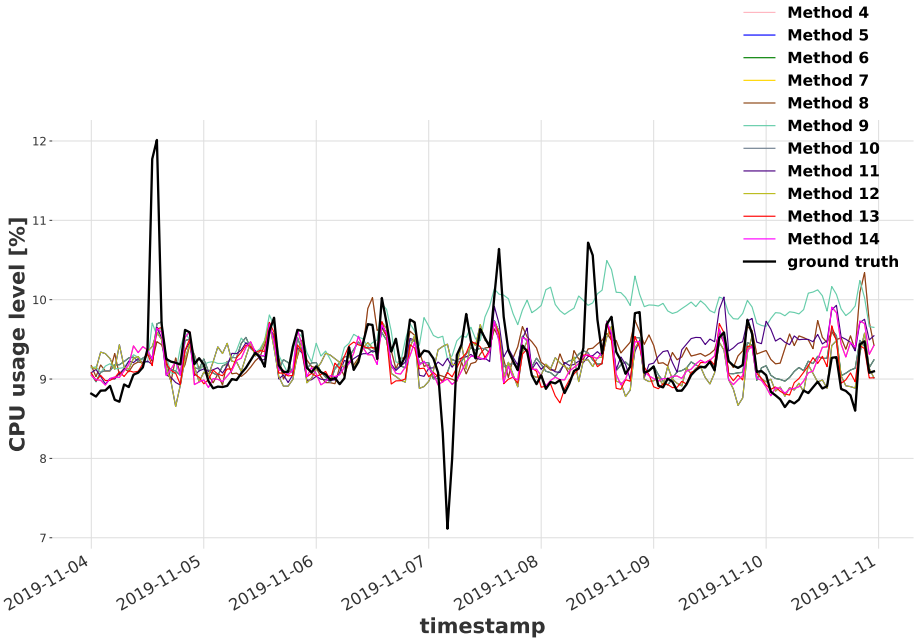

Fig. 4. CPU usage prediction results for XGBoost-based methods (first sample period)

As concerns the error metrics of methods that did not involve periodic adaptation to new data, it was found that anomaly detection within this specific dataset did not contribute to improved accuracy. The values identified by the anomaly detection module as outliers were found to have valuable positive impact. Therefore, *Method 4* outperformed *Methods 5*, *6* and *7* as removing anomalies led to the removal of valuable information and a dramatic reduction in accuracy. Moreover, the local applicability of anomaly detection also contributed to a significant accumulation of errors because scanning was applied without the broader context that was necessary for these data and for this model. *Method*

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

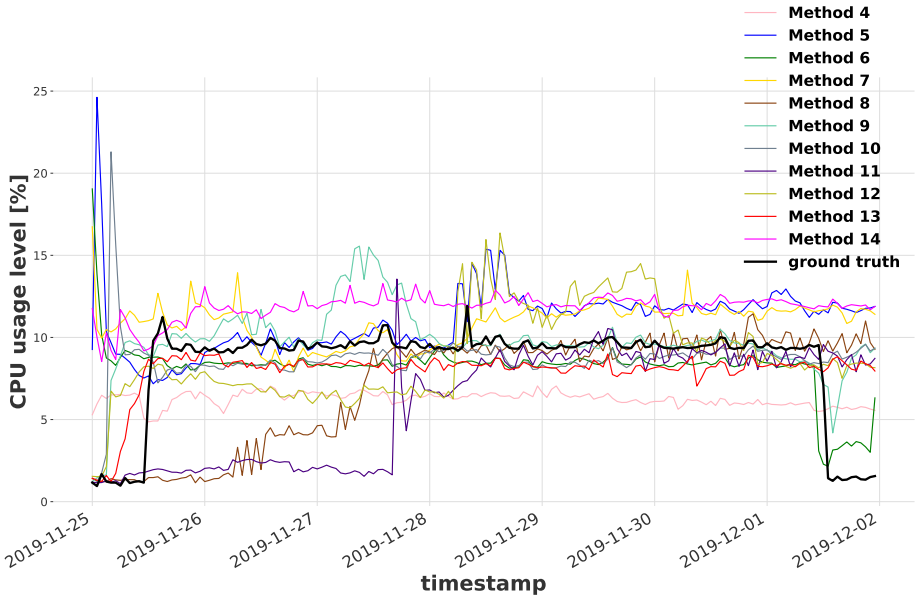

Fig. 5. CPU usage prediction results for XGBoost-based methods (second sample period)

Table 2. Error metrics for XGBoost-based methods obtained for the entire test set

Reference	RMSE	NRMSE	MAE	MAPE	MdAE
Method 4	2.477	0.157	1.355	0.502	0.407
Method 5	6.757	0.428	5.756	3.316	7.051
Method 6	14.762	0.936	9.352	5.121	1.882
Method 7	19.45	1.233	13.565	7.819	5.353
Method 8	1.814	0.115	0.909	0.335	0.335
Method 9	2.701	0.171	1.359	0.526	0.332
Method 10	2.573	0.163	1.185	0.58	0.321
Method 11	2.332	0.148	1.082	0.317	0.325
Method 12	2.627	0.167	1.195	0.546	0.321
Method 13	3.209	0.203	1.851	0.75	0.419
Method 14	3.209	0.192	1.399	0.646	0.334

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

8, which incorporated re-adaptation to new data and no anomaly detection, proved best in terms of RMSE, NRMSE and MAE metrics. The assessment of this method improved significantly because it uses walk forward validation. In a chaotic time series, validation error may sometimes be smaller than test error due to the variability of data distribution over time and differences between sample distributions among the training, validation and test sets. Chaotic time series are characterized by high variability and unpredictability, and models trained on training data may not be able to generalize correctly to new data. In this case, using a validation set may lead to a situation where the model is strictly optimized for validation data, which may represent only a small subset of test data. To prevent this, cross-validation techniques allow the model to be evaluated on multiple data subsets, reducing the risk of overfitting and allowing a more general model to be obtained. One advantage of this approach is that the model is trained on an increasing amount of data, which allows for continuous improvement over time. However, walk forward cross-validation in connection with grid search turned out be time-consuming, because it requires multiple iterations that involve training and testing the model on successive time windows. Therefore, it was important to carefully consider the appropriate validation method for the specific problem at hand. Refitting the model on an increasing amount of data enabled a better generalization to be to achieved. Additionally, all the subsequent methods incorporated this approach but also used anomaly detection. The results obtained using these methods confirmed that samples identified as anomalies in this particular time series have high information value and should not be discarded from the training set. However, this behavior should be constantly monitored as data characteristics may change over time. Generally, in the case of chaotic time series, in most cases the resource usage metrics recorded from HPC systems are in a form which makes continuous adaptation to new data useful, as data distribution is very dynamic and needs to be taken into account in order to maintain the accuracy of the predictive model.

5 Conclusions

Currently, cloud computing cannot fully replace supercomputers for applications such as advanced simulations using HPC, but it may play a complementary role, allowing calculations to be conducted where local resources are limited. In this work, we present a solution that enables long-term prediction of cloud resource consumption. A longer time horizon of resource consumption prediction is especially important when HPC is used for various simulations, which often run for a longer period of time. The proposed prediction system utilizes the XGBoost model and over a year of historical data to predict CPU resource usage. Experiments are conducted using real-life data from the production system. Their results indicate that proactive resource usage prediction, which models future usage and enables dynamic resource reservation in advance, is superior to reactive approaches. In the traditional approach, resources are reserved at their maximum capacity. In normal usage scenarios without prediction, reservation

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

remains at the highest level, even during periods of low demand. However, with the use of dynamic reservations, predictive capabilities are leveraged to scale resources dynamically in response to fluctuating demand.

It is worth noting that in our system, we have observed the superiority of the prediction system based on XGBoost over traditional statistical methods selected as the baselines. The best prediction method developed was found to outperform the best baseline model by approximately 16% in terms of RMSE. In addition to quantitative evaluation, the model exhibited significantly better qualitative responses to chaotic changes in data characteristics, suggesting that it is a more reliable predictor of system behavior. The output of the prediction system can subsequently be utilized to create a resource reservation plan, which can be used to scale resources.

Standard supervised learning operates under the assumption that each sample is drawn independently from an identical underlying distribution. However, chaotic time series and data from HPC systems are dynamic and often exhibit changes in data distribution, non-linearities, and non-stationarity, which makes accurate predictions difficult. Long-term prediction is a less explored area compared to the more prevalent short-term prediction, especially predicting one step ahead. However, the longer the prediction horizon, the less certain it becomes, but is at the same time more valuable, since it enables more informed decision-making, cost management and optimization of energy consumption. Setting a single universal rule is challenging, but evaluating multiple methods, especially using an incremental approach, allows strong conclusions to be drawn. In-depth exploratory data analysis may result in becoming more familiar with the data and putting forward solid predictions.

In further research, we would like to evaluate several models, such as LSTM (Long short-term memory Neural Network), TCN (Temporal Convolutional Network) and TFT (Temporal Fusion Transformer), to compare their effectiveness and conduct experiments using real HPC monitoring data. Additionally, we would like to explore multivariate prediction to incorporate dependencies between various HPC resource metrics, such as CPU, RAM, I/O, etc. Moreover, our objective is to dynamically evaluate the importance of features and reduce feature dimensionality by selecting a specific number of features for the model and summarizing the remaining features via Principal Component Analysis (PCA) or t-SNE. In addition, long-term prediction is of great significance and critical in HPC as it enables a balance between cost reduction and maintaining high-quality QoS by provisioning an appropriate amount of dynamically scalable resources rather than relying on statically reserved resources.

Acknowledgements The research presented in this paper was supported by funds from the Polish Ministry of Education and Science allocated to the AGH University of Krakow.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

References

1. Abhishek, M.K., Rajeswara Rao, D.: A scalable framework for high-performance computing with cloud. In: Tuba, M., Akashe, S., Joshi, A. (eds.) ICT Systems and Sustainability. pp. 225–236. Springer Nature Singapore, Singapore (2022)
2. Aljamal, R., El-Mousa, A., Jubair, F.: A user perspective overview of the top infrastructure as a service and high performance computing cloud service providers. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). pp. 244–249 (2019). https://doi.org/10.1109/JEEIT.2019.8717453
3. Chen, X., Wang, H., Ma, Y., Zheng, X., Guo, L.: Self-adaptive resource allocation for cloud-based software services based on iterative qos prediction model. Future Generation Computer Systems **105**, 287–296 (2020). https://doi.org/https://doi.org/10.1016/j.future.2019.12.005, https://www.sciencedirect.com/science/article/pii/S0167739X19302894
4. Garí, Y., Monge, D.A., Pacini, E., Mateos, C., García Garino, C.: Reinforcement learning-based application autoscaling in the cloud: A survey. Engineering Applications of Artificial Intelligence **102**, 104288 (2021). https://doi.org/https://doi.org/10.1016/j.engappai.2021.104288, https://www.sciencedirect.com/science/article/pii/S0952197621001354
5. Govindarajan, K., Kumar, V.S., Somasundaram, T.S.: A distributed cloud resource management framework for high-performance computing (hpc) applications. In: 2016 Eighth International Conference on Advanced Computing (ICoAC). pp. 1–6 (2017). https://doi.org/10.1109/ICoAC.2017.7951735
6. Gupta, A., Faraboschi, P., Gioachin, F., Kale, L.V., Kaufmann, R., Lee, B.S., March, V., Milojicic, D., Suen, C.H.: Evaluating and improving the performance and scheduling of hpc applications in cloud. IEEE Transactions on Cloud Computing **4**(3), 307–321 (2014)
7. Hazelhurst, S.: Scientific computing using virtual high-performance computing: A case study using the amazon elastic computing cloud. p. 94–103. SAICSIT '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1456659.1456671, https://doi.org/10.1145/1456659.1456671
8. Kotas, C., Naughton, T., Imam, N.: A comparison of amazon web services and microsoft azure cloud platforms for high performance computing. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–4 (2018). https://doi.org/10.1109/ICCE.2018.8326349
9. Mahmoudi, S.A., Belarbi, M.A., Mahmoudi, S., Belalem, G., Manneback, P.: Multimedia processing using deep learning technologies, high-performance computing cloud resources, and big data volumes. Concurrency and Computation: Practice and Experience **32**(17), e5699 (2020). https://doi.org/https://doi.org/10.1002/cpe.5699, https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5699
10. Nawrocki, P., Grzywacz, M., Sniezynski, B.: Adaptive resource planning for cloud-based services using machine learning. Journal of Parallel and Distributed Computing **152**, 88–97 (2021). https://doi.org/https://doi.org/10.1016/j.jpdc.2021.02.018, https://www.sciencedirect.com/science/article/pii/S0743731521000393
11. Nawrocki, P., Osypanka, P.: Cloud resource demand prediction using machine learning in the context of qos parameters. Journal of Grid Computing **19**(2), 20 (May 2021). https://doi.org/10.1007/s10723-021-09561-3

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53

12. Nawrocki, P., Osypanka, P., Posluszny, B.: Data-driven adaptive prediction of cloud resource usage. Journal of Grid Computing **21**(1), 6 (2023)
13. Nawrocki, P., Sus, W.: Anomaly detection in the context of long-term cloud resource usage planning. Knowledge and Information Systems **64**(10), 2689–2711 (2022)
14. Netto, M.A., Calheiros, R.N., Rodrigues, E.R., Cunha, R.L., Buyya, R.: Hpc cloud for scientific and business applications: taxonomy, vision, and research challenges. ACM Computing Surveys (CSUR) **51**(1), 1–29 (2018)
15. Osypanka, P., Nawrocki, P.: Qos-aware cloud resource prediction for computing services. IEEE Transactions on Services Computing pp. 1–1 (2022). https://doi.org/10.1109/TSC.2022.3164256
16. Osypanka, P., Nawrocki, P.: Resource usage cost optimization in cloud computing using machine learning. IEEE Transactions on Cloud Computing **10**(3), 2079–2089 (2022). https://doi.org/10.1109/TCC.2020.3015769
17. Posey, B., Deer, A., Gorman, W., July, V., Kanhere, N., Speck, D., Wilson, B., Apon, A.: On-demand urgent high performance computing utilizing the google cloud platform. In: 2019 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC). pp. 13–23. IEEE Computer Society, Los Alamitos, CA, USA (nov 2019). https://doi.org/10.1109/UrgentHPC49580.2019.00008, https://doi.ieeecomputersociety.org/10.1109/UrgentHPC49580.2019.00008
18. Radhika, E., Sudha Sadasivam, G.: A review on prediction based autoscaling techniques for heterogeneous applications in cloud environment. Materials Today: Proceedings **45**, 2793–2800 (2021). https://doi.org/https://doi.org/10.1016/j.matpr.2020.11.789, https://www.sciencedirect.com/science/article/pii/S2214785320394657, international Conference on Advances in Materials Research - 2019
19. Rahmanian, A.A., Ghobaei-Arani, M., Tofighy, S.: A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. Future Generation Computer Systems **79**, 54–71 (2018). https://doi.org/https://doi.org/10.1016/j.future.2017.09.049, https://www.sciencedirect.com/science/article/pii/S0167739X17309378
20. Singh, P., Gupta, P., Jyoti, K., Nayyar, A.: Research on auto-scaling of web applications in cloud: survey, trends and future directions. Scalable Computing: Practice and Experience **20**(2), 399–432 (2019)
21. Wittek, P., Rubio-Campillo, X.: Scalable agent-based modelling with cloud hpc resources for social simulations. In: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings. pp. 355–362. IEEE (2012)
22. Xue, S., Qu, C., Shi, X., Liao, C., Zhu, S., Tan, X., Ma, L., Wang, S., Wang, S., Hu, Y., Lei, L., Zheng, Y., Li, J., Zhang, J.: A meta reinforcement learning approach for predictive autoscaling in the cloud. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 4290–4299. KDD '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3534678.3539063, https://doi.org/10.1145/3534678.3539063

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_53