Predicting Compute Node Unavailability in HPC: A Graph-Based Machine Learning Approach

Roy Krumpak
Laboratory of Artificial Intelligence
Jožef Stefan Institute
Ljubljana, Slovenia
krumpak.roy@gmail.com

Jože M. Rožanec
Laboratory of Artificial Intelligence
Jožef Stefan Institute
Ljubljana, Slovenia
0000-0002-3665-639X

Martin Molan
DEI Department
University of Bologna
Bologna, Italy
martin.molan2@unibo.it

Matteo Angelinelli
HPC Department
CINECA
Bologna, Italy
0000-0002-3382-1900

Andrea Bartolini
DEI Department
University of Bologna
Bologna, Italy
a.bartolini@unibo.it

***Abstract*—As high-performance computing (HPC) systems advance towards Exascale computing, their size and complexity increase, introducing new maintenance challenges. Modern HPC systems feature data monitoring infrastructures that provide insights into the system's state. This data can be leveraged to train machine learning models to anticipate anomalies that require compute nodes to undergo maintenance procedures. This paper presents a novel approach to predicting such anomalies by creating a graph per measurement that encodes current and past sensor readings and information related to the compute node sensors. The experiments were performed with data collected from Marconi 100, a tier-0 production supercomputer at CINECA in Bologna, Italy. Our results show that the machine learning model can accurately predict anomalies and surpass current State-Of-The-Art (SOTA) models regarding the quality of predictions and the time horizon considered to forecast them.**

***Index Terms*—Artificial Intelligence, Machine Learning, Graphs, HPC, Data Center, Anomalies Forecasting**

I. Introduction

High-performance computing (HPC) is crucial for economic competitiveness, scientific leadership, and national security [1], [2]. Recognizing this, the European Commission launched its HPC strategy in 2012 and created the EuroHPC Joint Undertaking in 2018 [3]. Globally, similar efforts are advancing HPC capabilities, as seen in the Top500 list, which includes two exascale systems and eight pre-exascale systems [4]. While supercomputers lead in double precision FLOPs, generative artificial intelligence (AI) models like Meta's LLaMA 3.1 require even greater computational power in reduced precision, surpassing exaflop levels and, in some cases, requiring yottaflops of computational power [5].

As supercomputing performance increases, systems become more complex and prone to failures, especially in exascale clusters. A single-node failure can disrupt large-scale jobs, such as LLM training, if checkpointing is not in place. To address this, AI and data-driven methods, known as Operational Data Analytics (ODA), have been introduced alongside monitoring infrastructures [6]. The M100 Exadata [7], a public dataset from the Marconi100 cluster at CINECA, offers around 50TB of telemetry data, aiding in the analysis and research about managing large-scale HPC systems.

This work proposes a novel machine learning approach for predicting compute node anomalies (unavailability) in HPC systems. The models exploit two perspectives on data. First, we pre-process sensor signals to identify informative states and leverage their information to predict anomalies. Second, we create graph representations encoding domain knowledge and time series data at each point in time of a given compute node and leverage the graph embeddings to determine whether anomalies will take place in such compute nodes at a certain point in time in the future. We train and test our models on a subset of the abovementioned M100 Exadata dataset. The experimental results show our models likely surpass current SOTA models. Nevertheless, further work is required to fairly compare both models and draw definitive conclusions.

The rest of this paper is structured as follows: Section II describes the related work, Section III details the methodology we followed to train and test the machine learning models, and Section IV details the experiments we performed. The results we obtained are reported in Section V and briefly discussed VI. Finally, in Section VII, we present the conclusions and outline future work.

II. Related work

Operational Data Analytics in HPC domain ODA frameworks are essential for managing the complexity of modern HPC systems. They provide layers for data acquisition, processing, and visualization to support administrators handling large-scale systems. ODA frameworks, like Examon

This research was developed as part of the Graph-Massivizer project funded under the Horizon Europe research and innovation program of the European Union under grant agreement 101093202https://graph-massivizer.eu/ and supported by the Slovenian Research Agency.

Jože M. Rožanec and Roy Krumpak are co-first authors with equal contribution and importance

Corresponding author: Jože M. Rožanec: joze.rozanec@ijs.si

979-8-3503-5554-3/24/$31.00 ©2024 IEEE
DOI 10.1109/SCW63240.2024.00103

[8], collect real-time telemetry and log data. Integrated data-driven models such as PROCTOR [9], GRAAFE [10], and Wintermute [11] offer actionable insights, enhancing system monitoring and anomaly detection capabilities. This helps ensure more reliable operation and efficient management of HPC environments.

Anomaly detection and prediction are critical for the availability and sustainability of HPC systems, as noted by Netti et al. [12]. Anomalies, such as node failures, are periods where compute nodes are unavailable for jobs, as highlighted in the largest open HPC dataset [7]. While log-based methods like those of Tiwari [13] and Liu [14] predict component failures, node telemetry data is more commonly used due to privacy and performance concerns with log monitoring in large systems [15]. The GRAAFE model, utilizing graph neural networks, is the current state of the art, predicting anomalies up to six hours in advance [10], and serves as the benchmark for this paper's proposed method.

Graph methodologies for HPC monitoring GRAAFE [10], the current SOTA model, improves node anomaly prediction by incorporating the physical layout of compute nodes as a graph, where each node is a vertex connected to its neighbors, and telemetry data is represented as vertex attributes. A line graph topology representing a compute rack proved most effective, with a graph convolutional network predicting node failures in future windows through vertex classification. However, GRAAFE relies only on the last 15 minutes of telemetry data for prediction [7]. Research suggests that considering larger time windows enhances anomaly detection performance [16].

III. Methodology

Dataset The dataset we use in our research is a collection of sensor measurements gathered by Borghesi et al. from the Marconi 100 supercomputer [17] and made publicly available at https://zenodo.org/records/7541722. Given its petaflop computing capabilities, Marconi 100 corresponds to a Tier-0 European HPC facility. The system was co-designed by CINECA and replaced the former FERMI system in June 2016. Marconi 100 was replaced by Leonardo in 2023. For this research, we considered a fraction of the abovementioned dataset, the distribution file 1.tar, considering sensor measurements of seventeen compute nodes located in the racks of the supercomputer and taken between March 9th 2020 and September 28th 2022. The records do not provide raw sensor measurements but rather summarized values at fifteen-minute intervals.

Data preprocessing We only considered the averaged sensor values from the dataset, disregarding other available information, such as the variance of the values measured in the fifteen-minute intervals. Missing value imputation was performed with the Last Observation Carried Forward (LOCF) strategy, assuming that if the conditions in the compute node did not change much, the sensor values should remain close to the latest observed value. Furthermore, we observed that most sensor values remained close to a certain value for prolonged periods until a change level occurred. Therefore, we preprocessed the sensor data with a change-level detector. We simplified the data for each segment by replacing the actual sensor values with the average value of the whole segment. Furthermore, given the level changes were detected considering changes in the sensor mean values, we could still have sensor values that remained close to each other, signaling similar conditions and expected node behavior outcomes. We, therefore, decided to further simplify the sensor value representations by encoding them according to whether they belong to one of five quantiles for values observed for that particular sensor.

Identification of states prone to or that lead to anomalies After performing the pre-processing described above, we proceeded to identify states for each of the compute nodes, defining a state as a particular combination of sensor values observed at a certain point in time. Each state was given an ID. The states we obtained were used in two ways. First, we performed some clustering to identify which states were similar and had a higher density of anomalies. From this procedure, we found that three states entirely coincided with compute node anomalies. Second, we used the state IDs to re-encode the original time series into a sequence of state IDs and determine which states led to anomalous ones. With this procedure, we found that certain state sequences led to states associated with node downtimes.

Feature engineering As detailed above, we encoded sensor values according to the quantile they belonged to for each point in time. Therefore, we pursued two representations. First, we encoded the states as vectors, concatenating the one-hot encoded representations regarding which quantile a particular sensor value belonged at a certain time. Second, we created graphs describing the overall state of a node as sensed by the sensors per state change. The graphs are undirected and have a root node encoding the compute node ID. The root node edges lead to nodes describing sensor types. These nodes are linked to specific sensors of that type. Finally, attached to the sensor-specific nodes, we provided a natural visibility graph, considering the time series resulting from the last ten quantile states observed for that particular sensor. Natural visibility graphs [18] were considered given (i) the data was pre-processed into quantiles beforehand, providing a constrained set of time series values, (ii) they accurately capture the topology of a time series, and (iii) their encoding is positional, describing very well how transitions among last n states took place. To turn the graphs into features, we trained a Graph2Vec model and transformed each graph into an embedding of 15 values that could be used downstream to train machine learning models.

A. Model training

We considered two types of models: models trained only on data from a particular compute node (local models) and models trained on all available data (global models). Dividing the data following this criteria resulted in datasets with between 8.000 and nearly 12.000 instances for local models. We used a split of 75% of the data for training purposes, 5% for validation, and 20% for testing. Train, validation, and test sets were taken sequentially to preserve the order in which the sensor data was made available. In particular, the split dates considered were February 15th and April 1st 2022.

The models aimed to predict whether an anomaly (compute node unavailability) would take place at a specific node and time horizon. The time horizons were determined by state changes, which, on average, take place every 165 minutes. We considered three time horizons, predicting anomalies about 495 minutes (more than eight hours) ahead. We trained a CatBoost classifier for 250 iterations, considering a learning rate of 0.1 and an L2 regularization factor of 0.3 while using the cross-entropy loss and evaluating the models' performance on the validation set with a log-loss function.

B. Model evaluation

We evaluated the models' discriminative performance with the ROC AUC score. The score was computed at a compute node level and then summarized to report the average (AVG), minimum (MIN), and maximum (MAX) values obtained across nodes for each experiment.

IV. Experiments

We performed six experiments, training local and global models for three sets of feature vectors:

1) **one-hot encoded values** representing the quantile values to which each sensor reading belonged. This resulted in a feature vector of 66 values.
2) **Graph2Vec embeddings** under the assumption that random walks over the graph representation could be used to create a richer representation than the *one-hot encoded values*. This resulted in a feature vector of 15 values.
3) **joint representation** using *one-hot encoded values* and *Graph2Vec embeddings*, to validate whether they could provide complementary information to the model and lead to better results. This resulted in a feature vector of 81 values.

We refer to the experiments with IDs based on their feature sets and model types. E.g., 1G would refer to a *G*lobal model created with *one-hot encoded values*, while 3*L* would refer to a *L*ocal model created with features from a *joint representation*.

V. Results

In Table I, we present the results obtained for the six experiments described in the previous section. When comparing local and global models, we found that global models resulted in the best absolute average performance and led to the best performance ranges when considering compute node-specific forecasts. Only two exceptions were observed, with local models outperforming it when considering the minimum performance achieved by models predicting one state ahead and the maximum achieved performance of models predicting three states ahead. When considering feature sets, Graph2Vec embeddings displayed a very poor performance, showing ROC AUC values close to 0.5, with some exceptions reaching 0.6825. Nevertheless, when coupled to the *one-hot encoded values*, they strengthened the models' performance, leading to average ROC AUC values close to 0.85 when predicting one state ahead, with a slight decrease in performance when predicting two states ahead and with a significant performance drop (0.7886) when predicting three states ahead. Nevertheless, when considering the ranges of ROC AUC values achieved when forecasting anomalies for each of the compute nodes, performance was high as 0.9518 when forecasting one state ahead, achieving an even better score when forecasting two states ahead (0.9706), and achieving ROC AUC of 0.9060 when forecasting three states ahead. We consider the 3G model (global machine learning model trained with *joint representation* features) the best among the trained models. This model most likely significantly surpasses GRAFFE GNN, the current SOTA model. While the GRAFFE GNN achieves an ROC AUC between 0.78 and 0.91 with a four-hour look-ahead window, our 3G model achieves an ROC AUC performance between 0.7138 and 0.9785 for a look-ahead window of an average of 320 minutes (five hours and a half). Furthermore, when predicting three states ahead (an average of 495 minutes - eight hours and fifteen minutes), the worst-case performance decreases to 0.6549, and the best cases remain competitive, achieving an ROC AUC performance of 0.9060, while predicting more than twice ahead of the time horizon considered by GRAFFE GNN. While the results are promising, further work is required to strengthen the claims, such as comparing both models across the same test set splits.

VI. Discussion

While the work presented in this paper shows promising results, we must acknowledge certain limitations. First, results show the graph embeddings do a poor job on capturing information relevant to the classification task. This can be improved in many ways: trying different graph representations (e.g., encoding the physical layout of compute nodes within racks [19]), using different time series to graph encodings, graph embedding methods, and hyperparameter tuning. Our approach's advantages against GRAFFE is that the model is trained against meaningful state changes and not all sensor readings. This would potentially enable inference using real-time sensor data while not significantly increasing the dataset size to train the model. For example, such preprocessing reduces the effective dataset size about eleven times in the current dataset. While smaller datasets imply lower model training costs, no direct comparison against GRAFFE has been performed. We consider the proposed method to be generic. As long as data about sensor readings and metadata about sensors exist, it should be easily applied to other datasets and HPC environments to support exascale systems and enhance maintenance capabilities. Nevertheless, additional experiments are required to confirm this and remain a matter of future work.

VII. Conclusion

This paper presents a novel approach to predict compute node unavailability in HPC systems for Exascale computing.

Model type	Feature set	n+1			n+2			n+3		
		AVG	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**
Global	**1**	0.8315	0.6962	0.9281	**0.8475**	**0.6878**	0.9525	**0.7906**	**0.6707**	0.9077
	2	**0.4928**	**0.4065**	0.5802	0.5061	0.4024	0.6011	0.5219	**0.4080**	**0.6555**
	3	**0.8498**	**0.7147**	**0.9518**	**0.8485**	**0.7138**	**0.9785**	**0.7886**	**0.6549**	**0.9060**
Local	**1**	**0.8416**	**0.7311**	**0.9384**	0.8408	0.6658	**0.9706**	0.7903	0.6006	**0.9508**
	2	0.4908	0.3345	**0.5883**	**0.5285**	**0.4462**	**0.6825**	0.5342	0.3784	0.6671
	3	0.8335	0.6186	0.9313	0.8107	0.5568	0.9525	0.7806	0.5948	0.8970

TABLE I: ROC AUC values for machine learning models predicting anomalies occurrence for the Marconi 100 HPC system for time horizons of up to three states ahead. The best absolute results are underlined. The best results across feature sets are bolded.

The approach shows promising results and probably surpasses current SOTA models, achieving ROC AUC between 0.6549 and 0.9785 while predicting anomalies up to eight hours and a half ahead (more than twice the time horizon reported up to now for anomalies forecasting). Two factors contributed to the models' performance: (i) throughout sensor signal preprocessing to extract relevant information and remove noise, (ii) a simple representation of such states, and (iii) a hybrid graph representation of each state, combining time series information with domain knowledge regarding the sensors present in each compute node.

Future work will explore different graph representations to capture sensors' semantic attributes and better encode information about the time series. In addition, we will work closely with the authors of the GRAFFE GNN model to compare both models and provide a conclusive assessment of their performance.

References

[1] J. Dongarra, E. Deelman, T. Hey, S. Matsuoka, V. Sarakar, G. Bell, I. Foster, D. Keyes, D. Kranzlmueller, B. Lucas, et al., Can the united states maintain its leadership in high-performance computing?-a report from the ascac subcommittee on american competitiveness and innovation to the ascr office, Tech. rep., USDOE Office of Science (SC)(United States) (2023).

[2] F. Berberich, J. Liebmann, J.-P. Nominé, O. Pineda, P. Segers, V. Teodor, European hpc landscape, in: 2019 15th International Conference on eScience (eScience), IEEE, 2019, pp. 471–478.

[3] T. Skordas, Toward a european exascale ecosystem: the eurohpc joint undertaking, Communications of the ACM 62 (4) (2019) 70–70.

[4] J. J. Dongarra, H. W. Meuer, E. Strohmaier, 29th top500 Supercomputer Sites, Tech. rep., Top500.org (Nov. 1994).

[5] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models (2024). arXiv:2407.21783.
URL https://arxiv.org/abs/2407.21783

[6] M. Ott, W. Shin, et al., Global experiences with hpc operational data measurement, collection and analysis, in: 2020 IEEE International Conference on Cluster Computing, 2020.

[7] A. Borghesi, C. Di Santi, M. Molan, M. S. Ardebili, A. Mauri, M. Guarrasi, D. Galetti, M. Cestari, F. Barchi, L. Benini, F. Beneventi, A. Bartolini, M100 exadata: a data collection campaign on the cineca's marconi100 tier-0 supercomputer, Scientific Data 10 (1) (2023) 288. doi:10.1038/s41597-023-02174-3.
URL https://doi.org/10.1038/s41597-023-02174-3

[8] A. Borghesi, A. Burrello, A. Bartolini, Examon-x: a predictive maintenance framework for automatic monitoring in industrial iot systems, IEEE Internet of Things Journal (2021).

[9] B. Aksar, Y. Zhang, E. e. a. Ates, Proctor: A semi-supervised performance anomaly diagnosis framework for production hpc systems, in: High Performance Computing: 36th International Conference, ISC High Performance 2021, Virtual Event, June 24–July 2, 2021, Proceedings 36, Springer, 2021, pp. 195–214.

[10] M. Molan, M. S. Ardebili, J. A. Khan, F. Beneventi, D. Cesarini, A. Borghesi, A. Bartolini, Graafe: Graph anomaly anticipation framework for exascale hpc systems, Future Generation Computer Systems 160 (2024) 644–653. doi:https://doi.org/10.1016/j.future.2024.06.032.
URL https://www.sciencedirect.com/science/article/pii/S0167739X24003327

[11] A. Netti, M. Mueller, C. Guillén, M. Ott, D. Tafani, G. Ozer, M. Schulz, DCDB wintermute: Enabling online and holistic operational data analytics on HPC systems, CoRR abs/1910.06156 (2019). arXiv:1910.06156.
URL http://arxiv.org/abs/1910.06156

[12] A. Netti, W. Shin, M. Ott, T. Wilde, N. Bates, A conceptual framework for hpc operational data analytics, in: 2021 IEEE International Conference on Cluster Computing (CLUSTER), 2021, pp. 596–603. doi:10.1109/Cluster48925.2021.00086.

[13] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, W. Shi, Making disk failure predictions smarter!, in: Proceedings of the 18th USENIX Conference on File and Storage Technologies, FAST'20, USENIX Association, USA, 2020, p. 151–168.

[14] Y. Liu, Y. Guan, T. Jiang, K. Zhou, H. Wang, G. Hu, J. Zhang, W. Fang, Z. Cheng, P. Huang, Spae: Lifelong disk failure prediction via end-to-end gan-based anomaly detection with ensemble update, Future Generation Computer Systems 148 (2023) 460–471. doi:https://doi.org/10.1016/j.future.2023.05.020.
URL https://www.sciencedirect.com/science/article/pii/S0167739X23002030

[15] P. Matri, P. Carns, R. Ross, A. Costan, M. S. Pérez, G. Antoniu, Slog: Large-scale logging middleware for hpc and big data convergence, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2018, pp. 1507–1512.

[16] M. Molan, A. Borghesi, D. Cesarini, L. Benini, A. Bartolini, Ruad: Unsupervised anomaly detection in hpc systems, Future Generation Computer Systems 141 (2023) 542–554. doi:https://doi.org/10.1016/j.future.2022.12.001.

[17] A. Borghesi, C. Di Santi, M. Molan, M. S. Ardebili, A. Mauri, M. Guarrasi, D. Galetti, M. Cestari, F. Barchi, L. Benini, et al., M100 exadata: a data collection campaign on the cineca's marconi100 tier-0 supercomputer, Scientific Data 10 (1) (2023) 288.

[18] V. F. Silva, M. E. Silva, P. Ribeiro, F. Silva, Time series analysis via network science: Concepts and algorithms, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11 (3) (2021) e1404.

[19] M. Molan, J. Ahmed Khan, A. Borghesi, A. Bartolini, Graph neural networks for anomaly anticipation in hpc systems, in: Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, 2023, pp. 239–244.