Turkish Journal of Computer and Mathematics Education (TURCOMAT) ISSN: 3048-4855

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

OPTIMIZING NUMERICAL WEATHER PREDICTION MODEL PERFORMANCE USING MACHINE LEARNING TECHNIQUES

[1] *Dr K Venkata Naganjaneyulu*,[2] *N. Nayana Latha*,[3] *P. Jhansi Sriya*,[4] *P. Vaishnavi*
[1]*Professor*,[234]*Students*
Department Of CSE
Malla Reddy Engineering College for Women

ABSTRACT

Numerical weather prediction models, which use weather observation data such as temperature and humidity, are the main tool used in weather forecasting. For weather forecasting, the UK-based GloSea6 numerical weather prediction model has been used by the Korea Meteorological Administration (KMA). Supercomputers are necessary to run these models for research reasons in addition to using them for real-time weather predictions. However, several researchers have encountered challenges in running the models because of the restricted supercomputer resources. In order to solve this problem, the KMA created the Low GloSea6 low-resolution model. Although Low GloSea6 can operate on small and medium-sized servers in research facilities, it still consumes a lot of computer resources, particularly in the I/O load. Model I/O optimization is crucial as I/O load may lead to performance deterioration for models with heavy data I/O; nevertheless, user trial-and-error optimization is ineffective. In order to improve the hardware and software characteristics of the Low GloSea6 research environment, this work provides a machine learning-based method. There were two stages in the suggested procedure. In order to extract hardware platform parameters and Low GloSea6 internal parameters under different settings, performance data were first gathered using profiling tools. Second, the acquired data was used to build a machine learning model that identified the ideal hardware platform parameters and Low GloSea6 internal parameters for fresh study settings. When compared to the actual parameter combinations, the machine-learning model demonstrated a high degree of accuracy in its successful prediction of the ideal combinations of parameters in various research situations. Specifically, a noteworthy result was the error rate of just 16% between the actual execution time and the expected model execution time based on the parameter combination. All things considered, this optimization technique may enhance the efficiency of further high-performance computing research applications.

1. INTRODUCTION

Due to substantial advancements in computing power, numerical weather prediction (NWP) [1] techniques that rely

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

heavily on numerical simulations to predict the weather have been more popular. In 1999, the Korea Meteorological Administration (KMA) began using a worldwide data assimilation and forecasting system that relied on the global spectral model. This model was derived from the global spectrum model used by the Japan Meteorological Agency. The UK Met Office's global NWP model GloSea6 [2] started being used by the KMA for weather forecasting in 2022.

GloSea6 primarily uses the ATMOS and OCEAN models. In contrast to the OCEAN model, which include both ocean and sea ice components, the ATMOS model separates these two components into atmospheric and land surface components, respectively. After the Earth is divided into grids and starting and auxiliary data, called analysis fields, are collected for each grid during the preprocessing phase, the model is run. After the analytical fields are prepared for use in the forecast model, the numerical model computing procedure may begin.

The KMA provides a low-resolution version of GloSea6, called Low GloSea6, for researchers who do not have access to supercomputers because of the large processing power consumption. Due to the high volume of data inputs and outputs (I/O) required by the model, even Low GloSea6 requires substantial computing power. It is worth mentioning that typical consumers, who are more likely to be researchers in atmospheric science than computer scientists, may see trial-and-error speed improvement as inefficient. This work introduces a machine learning-based approach to enhance the Low GloSea6 research environment's hardware and software features.

An innovative cross-inference optimization method for the Low GloSea6 NWP model is presented in this work by combining benchmark tools with machine learning. This pertains specifically to the following details:
Everything about the performance cross-validation approach was something we came up with and tested.
The data needed for cross-inference was separated into two categories: the execution hardware platform parameters and the internal software parameters of Low GloSea6. The validation of the models and the data allowed us to retrieve the critical parameters for each group.
• After collecting extensive data on I/O characteristics using Darshan, we confirmed our final conclusions using I/O performance cross-validation using runtime data. • This study demonstrates the applicability of various machine-learning techniques to explain the intricate interactions between the Low GloSea6 internal software parameters and the execution hardware platform

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

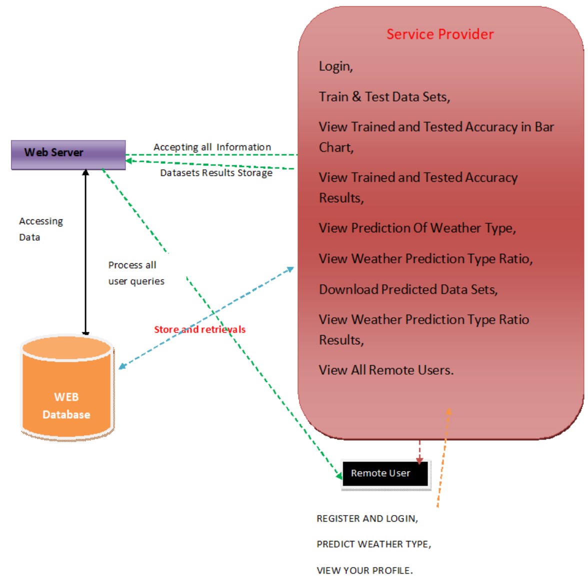
parameters, making it possible to cross-infer performance on a new execution hardware platform.

• This research's focus, Low GloSea6, is not unique to the proposed method; the methodology has been expanded to demonstrate this.

Here is the format of the document: Section II discusses relevant research, while Section III provides an in-depth analysis of GloSea6, a numerical model for weather prediction and the profiling tool used to get performance data. In Section IV, we detail the model, dataset, and hardware/software optimization approach employed in the research setting. Section V delves into the testing procedures that followed the optimization-based model and data verification process. In Section VI, we lay out our last thoughts and future plans.

2. SYSTEM ARCHITECTURE

Architecture Diagram



3. EXISTING SYSTEM

Applications in both academic and industrial settings have been the focus of optimization studies in many different fields. Application I/O optimization may be accomplished, for example, by modifying the codes of I/O libraries. Howison et al. [3] shown how to improve performance for HPC applications by enhancing the MPI-IO and HDF5 libraries and making code modifications that take file system quirks into consideration. Finding the best file systems and I/O library properties is another research strategy for optimizing I/O. Plus, Behzad et al. [4], [5] used genetic optimization to improve the I/O performance of an application. Their collection of parameters was based on their investigation of the file system and I/O

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

library's parameter space. They tested the benchmark tool's I/O performance using the given parameters and adjusted them iteratively based on the results until they got the best possible I/O performance.

Robert et al. [6] improved an I/O accelerator using black-box optimization techniques, which find input parameters with greatest and lowest performance metrics without considering internal operations. The Atos Flash Accelerator is an I/O accelerator that utilizes NAND flash memory technology to speed up I/O operations of different HPC applications. They optimized three input parameters—I/O throughput, I/O latency, and I/O memory usage—using fundamental metrics like I/O operation processing time as performance indicators.

Finally, they confirmed that the I/O accelerator's performance could be improved via black-box optimization. An automated tuning approach for the optimal configurations of the Lustre parallel file system, a high-performance variant of MPI-IO, and the MPI-IO ROMIO library was developed by Bağbaba et al. [7] using I/O monitoring and performance prediction. The outcome was achieved by the use of a random forest-based machine learning strategy and a molecular dynamics model (ls1 Mardyn). Two benchmarking tools, MPI-Tile-IO and IOR-IO, confirmed it. Our research differs from previous findings in two key respects.

The first thing to know is that our research simplifies optimization, even for those who have never done it before. Changing the I/O library code increased I/O performance (Howison et al., 2003), however this solution requires developer skill and is difficult for non-developers to use. In contrast, we optimize performance using machine learning, which can be easily accessible and adjusted by considering the software and hardware components of the study environment. Secondly, our study takes into consideration both the internal software characteristics and the platform specifications of the hardware all at the same time.

With a focus on file systems, HDF5, and MPI-IO libraries, Behzad et al. [4], [5] improved I/O using configurable parameters in the parallel I/O stack. However, the research did not include optimizing the parameters of the benchmark tool. Using the I/O latency, I/O throughput, and I/O memory utilization characteristics of the Atos Flash Accelerator I/O accelerator, Robert et al. [6] optimized the system. This research only addresses these internal software settings and makes no mention of the platform's hardware specifications.

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

Numerous fields may benefit from our research. Since Bağbaba et al. [7] only looked at the Lustre parallel file system and the MPI-IO ROMIO library in a specific research environment, their findings may not be applicable to other situations. Conversely, we made advantage of Low GloSea6 to confirm our findings in two other research settings and hardware platform types. And we utilized MPICH, an easily accessible implementation of MPIIO that works regardless of the version of MPICH itself. We conducted our testing using many MPICH versions in research conditions to verify this.

Disadvantages

- ➢ The classic bagging approach is enhanced with a sequential feature in the gradient boosting model, which does not use weights to convert weak models into strong ones.
- ➢ The MLR model, which is typical of linear regression estimation, does not use hyperparameters. A hyperparameter known as "mtry" in the random forest model controls the amount of features used by each tree.

4. PROPOSED SYSTEM

Using machine learning and benchmark tools, the system proposes a new cross-inference optimization method for the Low GloSea6 NWP model. This pertains specifically to the following details:

Everything about the performance cross-validation approach was something we came up with and tested. The data needed for cross-inference was separated into two categories: the execution hardware platform parameters and the internal software parameters of Low GloSea6. Model and data validation helped to extract critical parameters from each category.

• After collecting extensive data on I/O characteristics using Darshan, we confirmed the final results using I/O performance cross-validation using runtime data.

Thanks to this study, we can now cross-infer performance on a new execution hardware platform by understanding the complex interactions between the execution hardware platform parameters and the Low GloSea6 internal software parameters through the application of various machine-learning techniques.

• The procedure has been extended to demonstrate that the proposed method is an all-encompassing strategy unrelated to the Low GloSea6 case study.

Advantages

- ➢ We provide a new cross-inference optimization approach that uses machine learning and benchmark tools to improve Low GloSea6's performance by considering the platform's hardware as well as the application's internal characteristics.

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

- Predicting the dependent variable using the MLR approach is based on the premise that the independent and dependent variables are linearly related. Random forest and gradient boosting are two ensemble techniques that use decision trees. To compensate for decision trees' instability, the ensemble technique mixes weak models to create a strong model.

5. IMPLEMENTATION

Modules description

Service Provider

To access this module, the Service Provider has to provide a valid username and password. Upon successful login, he will be able to access certain activities, such as Train and Test Data Sets and Examine the Bar Chart for Trained and Tested Accuracy. You may see all remote users, access trained and tested accuracy results, download predicted data sets, and see the weather prediction type ratio and prediction.

View and Authorize Users

All users who have registered for this module may be seen by the administrator. Users' names, email addresses, and physical addresses are viewable to the administrator, who may also authorize users.

Remote User

In this module, you will find n users. The user must register before they may begin any activity. Upon registration, the user's details are stored in the database. He will be prompted to enter his approved username and password after he successfully registers. After logging in, users have a lot of options, such as seeing their profile, making weather predictions, and registering and logging in again.

6. CONCLUSION

This research revealed a machine learning-based method for improving the hardware and software characteristics of scientific applications. Targeted was the scientific weather forecasting program Low GloSea6, for which a dataset including hardware platform parameters, application internal parameters, and performance data derived from the combination of these two elements was created. The dataset was checked before the machine-learning model was applied, and the LOOCV approach was used to guarantee the correctness of the regression model developed on inadequate data. Using the trained machine-learning model in a fresh study setting, the ideal hardware platform parameters and matching Low GloSea6 internal parameters were discovered, and these values coincided with the actual parameter combinations. Specifically, a significant outcome in terms of execution time prediction was shown

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

when the estimated execution time based on the parameter combination revealed a 16% error rate in comparison to the actual execution time. Other HPC scientific applications may perform better when using the suggested optimization technique. In addition to climate and weather modeling, other examples include quantum chemistry computations, molecular dynamics (MD) simulations, and computational fluid dynamics (CFD) simulations. Our optimization approach will help speed up the manual performance optimization procedure that scientists running these kinds of HPC research applications used to receive assistance from supercomputing center workers to improve their programs.

Data-wise, two avenues for further investigation are indicated. First, there must be an increase in the total quantity of data. In this investigation, the absence of certain hardware platform characteristics made it more difficult to forecast execution time accurately. Therefore, the performance of the model would be enhanced by gathering more hardware/software parameters and I/O performance indicators. Second, it would be advantageous to put into practice the benchmark-based cross-inference optimization strategy that was first suggested in this study's algorithm. By using other parameters, this would allow for the gathering of parameter values not obtained in this research, which would speed up data collection and increase the model's application range and performance.

REFERENCES

[1] *Concept of a Numerical Forecast Model*. Accessed: Aug. 10, 2023. [Online]. Available: http://web.kma.go.kr/aboutkma/intro/supercom/ model/model_concept.jsp

[2] P. Davis, C. Ruth, A. A. Scaife, and J. Kettleborough, ‘‘A large ensemble seasonal forecasting system: GloSea6,’’ Dec. 2020, vol. 2020.

[3] M. Howison, Q. Koziol, D. Knaak, J. Mainzer, and J. Shalf, ‘‘Tuning HDF5 for Lustre file systems,’’ Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-4803E, 2010. [4] B. Behzad et al., ‘‘Taming parallel I/O complexity with auto-tuning,’’ in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2013, p. 68.

[5] B. Behzad, S. Byna, Prabhat, and M. Snir, ‘‘Optimizing I/O performance of HPC applications with autotuning,’’ *ACM Trans. Parallel Comput.*, vol. 5, no. 4, pp. 1–27, Mar. 2019, doi: 10.1145/3309205.

[6] S. Robert, S. Zertal, and G. Goret, ‘‘Auto-tuning of IO accelerators using black-box optimization,’’ in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2019, pp. 1022–1027, doi: 10.1109/HPCS48598.2019.9188173.

Vol. 15 No. 3 (2024):278-285
DOI: https://doi.org/10.61841/turcomat.v15i3.14801

[7] A. Bağbaba, X. Wang, C. Niethammer, and J. Gracia, ‘‘Improving the I/O performance of applications with predictive modeling based auto-tuning,’’ in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Oct. 2021, pp. 1–6, doi: 10.1109/ICEET53442.2021.9659711.

[8] S. Valcke and R. Redler, ‘‘The OASIS coupler,’’ in *Earth System Modelling*, vol. 3. Berlin, Germany: Springer, 2012, pp. 23–32, doi: 10.1007/978-3-642-23360-9_4.

[9] *Analysing UM Outputs*. Accessed: Feb. 14, 2023. [Online]. Available: http://climate-cms.wikis.unsw.edu.au/Analysing_UM_outputs

[10] *Unidata | NetCDF*. Accessed: Nov. 28, 2022. [Online]. Available: https://www.unidata.ucar.edu/software/netcdf/

[11] Icons8. *Free Icons, Clipart Illustrations, Photos, and Music*. Accessed: Jul. 18, 2023. [Online]. Available: https://icons8.com

[12] P. Carns, R. Latham, R. Ross, K. Iskra, S. Lang, and K. Riley, ‘‘24/7 characterization of petascale I/O workloads,’’ in *Proc. 2009 Workshop Interfaces Archit. Sci. Data Storage*, Sep. 2009, pp. 1–10.

[13] *Darshan Introduction*. Accessed: Aug. 10, 2023. [Online]. Available: https://wordpress.cels.anl.gov/darshan/wp-content/uploads/sites/54/ 2014/08/iiswc-2014-darshan-instrumentation.pdf

[14] R. Ross, D. Nurmi, A. Cheng, and M. Zingale, ‘‘A case study in application I/O on Linux clusters,’’ in *Proc. ACM/IEEE Conf. Supercomput.*, New York, NY, USA, Nov. 2001, p. 11, doi: 10.1145/582034.582045.

[15] S. Herbein, D. H. Ahn, D. Lipari, T. R. W. Scogland, M. Stearman, M. Grondona, J. Garlick, B. Springmeyer, and M. Taufer, ‘‘Scalable I/Oaware job scheduling for burst buffer enabled HPC clusters,’’ in *Proc. 25th ACM Int. Symp. High-Perform. Parallel Distrib. Comput.*, New York, NY, USA, May 2016, pp. 69–80, doi: 10.1145/2907294.2907316.