A novel prediction model for combined computational and storage malleability

Paula Sanchez-Checa
paulasan@inf.uc3m.es
University Carlos III of Madrid
Leganés, Madrid, Spain

David E. Singh
dexposit@inf.uc3m.es
University Carlos III of Madrid
Leganés, Madrid, Spain

Javier Garcia-Blas
fjblas@inf.uc3m.es
University Carlos III of Madrid
Leganés, Madrid, Spain

Jesus Carretero
jcarrete@inf.uc3m.es
University Carlos III of Madrid
Leganés, Madrid, Spain

ABSTRACT

Malleability enables to dynamically adapt the parallel applications' resources during program execution. These modifications are made based on different performance criteria. If needed and available, new resources are added to improve execution time and meet the desired performance. Likewise, resources are released if they are not needed, as having more resources than needed may result in an increase of communication time and so increasing the application's execution time. Additionally, other applications running on the same cluster can benefit from the released resources. The aim of this work is to dynamically predict the I/O requirements of the application based on its computation requirements and by predecessor time window traces. The predictions are based on temporal series with sliding time windows. The main objective is to adapt the number of data nodes depending on the application's I/O requirements. In this way, the storage system will be deployed the moment the application needs it, avoiding overloads and slowdowns.

KEYWORDS

Malleability, High Performance Computing, Storage

ACM Reference Format:
Paula Sanchez-Checa, David E. Singh, Javier Garcia-Blas, and Jesus Carretero. 2024. A novel prediction model for combined computational and storage malleability . In *Proceedings of (womENcourage).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Historically, the primary focus of HPC system providers and users has been enhancing the parallel computing performance. However, the burgeoning demands for data processing in emerging applications like machine learning are compelling all parties involved to reassess their perspectives on the HPC landscape. Striking a balance between the computational and I/O demands of these applications could optimize the utilization of existing HPC resources and enhance overall performance.

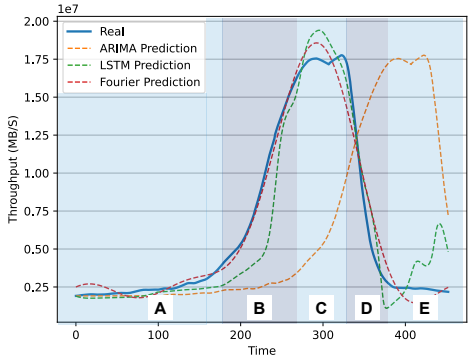

Figure 1: Real and predicted values of ARIMA, LSTM and Fourier transforms with a 100 time step window size.

EpiGraph [5] is a parallel application based on MPI that simulates the propagation of epidemic diseases (i.e., COVID-19) using epidemic and social models related to large urban areas. The current implementation of EpiGraph is malleable, which means that the application is able to expand or shrink the number of processes during its execution. On the other hand, Hercules [2] is an in-memory ad-hoc file system that aims to improve I/O throughput by tuning multiple parameters such as the network topology, network transfer size, and others.

This work focuses on improving the I/O performance of EpiGraph simulator when it is used in combination with Hercules file system. Similarly as with processing, I/O requirements may change during runtime. Thus, the main idea behind this work is to use the performance model of EpiGraph for predicting the future application I/O throughput. In addition, the good use of computing resources contributes to execution energy efficiency, advocating for responsible computing.

Figure 1 shows a trace of an EpiGraph simulation. This trace can be divided into five different stages, A through E, based on the evolution of the I/O throughput. Stage A is related to the beginning of the execution, when there are not many active infections. In this

womENcourage, June 26–28, 2024, Madrid, ES

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

case the iteration execution time is reduced, thus the I/O operations will be performed more frequently, producing a high I/O demand. In stage B, the I/O throughput starts to reduce because of an increase in the number of infections. This produces an increase in the iteration execution time that spaces the I/O operations over time. In stage C, the simulation reaches the peak in the curve of infections (large number of computation with maximum CPU times), reaching the minimum I/O throughput values. In stage D, the throughput starts to increase again due to the decrease in the infection wave. Finally, in stage E, the simulation only considers a marginal number of infections, and a similar throughput to the one we had at the beginning of the execution is reached.

In this scenario, our work gains significance, as the data nodes should adapt to the situation in order to avoid bottlenecks and slowdowns in the execution time. Note that with this strategy we will be able to reduce the number of resources allocated by Hercules during the simulation execution (i.e., memory footprint).

Figure 2 depicts the proposed architecture for deploying a malleable file system. The computing architecture is represented over the green rectangle and the persistent storage using the blue one. It is formed by a collection of compute and storage nodes, represented in purple and blue respectively. Nodes displayed with a solid line represent nodes that are being used by the application, while nodes represented with a dashed line represent nodes that are available and can be allocated if necessary.

The system software components are displayed on the right of the figure. The resource management is in charge of applying malleability to EpiGraph (Arrow 1), using FlexMPI [4], expanding or shrinking the application depending on the application's needs. In Figure 2, the computing nodes are expanded from two to four nodes. Arrow 2 represents the monitoring phase, which obtains information about EpiGraph's computation nodes and Hercules I/O nodes. The time-series model uses the data obtained by the monitoring phase to predict the number of data nodes that are required by the application. The decision-making phase is in charge of deciding whether or not it is necessary to expand the number of data nodes of Hercules file system. Modifications are predicted in advance, and the data nodes will be ready to expand or shrink before the I/O requirements change. In the example shown in Figure 2, the model decides to expand one data node through B. Note that the number of nodes used by Hercules (three in this example) does not have to be the same as the ones used by EpiGraph (four in the example), given that the I/O operations are performed by network communications.

2 METHODOLOGY

We have studied three different types of time-series based models for predicting I/O requirements. On one hand we have studied the ARIMA model, a stochastic model used for predicting the future values of a variable based on its historical data. This model is frequently used for time series prediction, and assumes that the future values of a variable are a linear combination of the previous values [1]. We have also studied LSTM, a type of recurrent neural network that shows great performance when predicting long-term data [3]. Finally, we have studied Fourier transforms, which are traditionally used for signal processing to map our time sampled data to the

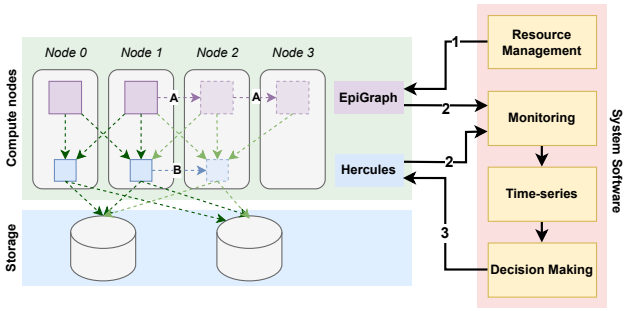

Figure 2: Architectural design of the malleable EpiGraph/Hercules application.

frequency domain. This has allowed us to clean data noise and obtain a fitting for the throughput traces.

Figure 1 shows a part of a real epigraph trace together with the 100 time steps ahead predictions obtained by the three models that have been studied. It can be observed that while LSTM and Fourier methods obtain good approximations of the trace, ARIMA method shows a lag in the predictions.

3 CONCLUSIONS AND FUTURE WORK

We have studied the performance of different methods in predicting Epigraph's throughput. While the results obtained for ARIMA method are not as accurate as we need, LSTM neural networks and Fourier transform methods have shown promising results in throughput forecasting. The use of these models to predict I/O demands can be a key point of improvement in the performance of the application.

So far, our work has focused on forecasting the applications throughput when executed with a specific number of processes. To continue our research, we would further look at how our methods can be adapted to a number of changing processes in the application. In addition, we would like to test the improvement of EpiGraph's performance when applying the above presented methodology.

ACKNOWLEDGMENTS

This work has been partially funded by the European High-Performance Computing Joint Undertaking (JU) under the ADMIRE project (grant agreement No 956748) and the Spanish Research Agency under grant PCI2021-121966.

REFERENCES

[1] K. Lalitha Devi and S. Valli. 2023. Time series-based workload prediction using the statistical hybrid model for the cloud environment. *Computing* 105, 2 (Feb 1, 2023), 353–374. https://doi.org/10.1007/s00607-022-01129-7

[2] Javier Garcia-Blas, Genaro Sanchez-Gallegos, Cosmin Petre, Alberto Riccardo Martinelli, Marco Aldinucci, and Jesus Carretero. 2023. Hercules: Scalable and Network Portable In-Memory Ad-Hoc File System for Data-Centric and High-Performance Applications. In *Euro-Par 2023*. Cyprus, 679–693.

[3] Joos Korstanje and Michael Keith. 2021. *Advanced Forecasting with Python : With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR* (1 ed.). Apress, Netherlands. https://doi.org/10.1007/978-1-4842-7150-6

[4] Gonzalo Martín, David E. Singh, Maria-Cristina Marinescu, and Jesús Carretero. 2015. Enhancing the performance of malleable MPI applications by using performance-aware dynamic reconfiguration. *Parallel Comput.* 46 (2015), 60–77.

[5] Miguel Guzman Merino, Maria-Cristina Marinescu, Alberto Cascajo, Jesus Carretero, and David E. Singh. 2023. Evaluating the spread of Omicron COVID-19 variant in Spain. *Future Generation Computer Systems* 149 (2023), 547–561.