IMTC 2003 – Instrumentation and Measurement
Technology Conference
Vail, CO, USA, 20-22 May 2003

Frequent patterns-based prefetching and caching for improving the performance of cloud storage systems

Ms. Anusha Nalajala[1], Kolli Deepthi[2], Mediboena Lavanya Deepthi[2], Anusha Kandula[2] and Vamsi Kishore Nallagopu[2]
Department of Computer Science and Engineering, SRM University - AP
Neerukonda, Mangalgiri, Guntur, Andhra Pradesh
Phone: +91 9052017817, Email: anusha.n@srmap.edu.in

Abstract *– The vast quantities of data generated by data-intensive applications are most effectively stored using distributed file systems (DFS) in a cloud computing environment. This research presents a graph-based frequent pattern generation approach with machine learning optimization to optimize read operations in DFS environments. Frequent patterns are generated by graph-based approach that capture data relationships, these patterns can be used to gain high-level analysis, fostering knowledge discovery and decision support. Developed a specialized prefetching technique using LSTM model to enable adaptability to different system configurations. This model succeeds in finding frequent patterns of relevance and usage trends that affect system performance optimization. By using the LSTM model for predicting frequent patterns, these forecasted patterns are afterward used as guidance for prefetching and DFS processes, thereby improving proficiency and immediate responsiveness at the level of reading. ML-driven prefetching techniques integrated with DFS operations represent a remarkable breakthrough in data storage and retrieval approaches that ensure maximized performance and scalability in data-driven environments. Through extensive experimentation and performance evaluation, the proposed methodology demonstrates significant reductions in average read access time compared to traditional methodologies, highlighting its efficacy in improving DFS performance for data-intensive applications.*

Keywords- Distributed File System, LSTM, Frequent patterns, prefetching and data-intensive application

INTRODUCTION

Modern web servers routinely generate data, known as weblogs, as users interact with websites. These logs contain various details such as User ID, Application ID, Rack ID, Data Node ID, Block ID, File ID, and timestamp. Rack-organized architecture integrated to leverages the Distributed File System (DFS) concept, . Within this architecture, physical servers are grouped into racks, each housing multiple Data Nodes tasked with data block storage and management. Each Data Node can accommodate blocks from different files, with replication mechanisms ensuring data redundancy and availability. Files are segmented into smaller blocks, typically ranging from tens to megabytes in size. These blocks are then distributed across multiple Data Nodes, enhancing fault tolerance, reliability, and load balancing. Refer to Figure-1 for an illustration of the DFS architecture.

The aim is to solve the problems related to performance optimization and the read operations for DFS systems with a graph-based algorithm. Graphs are used to trace interactions between data sets and the frequency of their usage, which helps in identifying the factors that affect data access behaviors. The machine learning model like Long Short-Term Memory (LSTM) is used for further optimization. Previous data access sequences are used to train the LSTM model for predicting frequent patterns which helps in reducing latency and enhancing read operation efficiently. The research provides insights to gain practical experience in technologies such as graph theory and machine learning along with providing optimization of DFS performance for real-world applications in the era of big data.

A. Motivation

In this age of big data, it has become increasingly critical to understand user behavior on the web for various purposes like personalized recommendation systems, targeted advertising, and user experience optimization. Efficiently managing and analyzing vast amounts of data has become crucial. Distributed file systems (DFS) are utilized to store and retrieve data and enable the smooth operation of data-intensive applications across various domains. However, to maintain huge volumes of data traditional DFS systems frequently struggle to maintain ideal performance levels, particularly in read operations. Researchers have proposed several approaches to analyze and mine patterns from web server logs to reveal useful insights concerning user browsing patterns for read-time optimization.

B. Scope

Real-world weblog data, such as those from Amazon, anticipates user needs by analyzing browsing habits. To know how it proactively stores frequently accessed data, such as product pages and images, latency is reduced, increasing user satisfaction and system efficiency this research work was intiated. This increases engagement and loyalty, thereby improving the overall customer experience.

C. Objective

The primary goal of this proposed methodology is frequent pattern-based prefetching and caching. Initially, the proposed methodology will involve an extensive analysis of weblogs within the cloud storage system to identify frequent patterns. This analysis delves deeply into user behaviour, examining which files or data are accessed frequently, how often, and in what order. The proposed goal is to identify involved usage patterns that can be used to develop predictive algorithms.

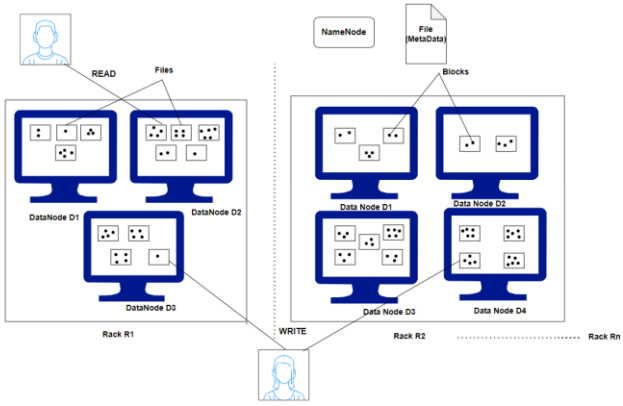

Fig. 1. DFS Architecture

RELATED WORKS

In the digital space, data retrieval management, and optimization are the two emerging focal points in information technology. Researchers are looking for solutions that would facilitate and ensure the accuracy of data collection. They employ technologies focusing on understanding user behaviour, especially their web browsing patterns. This underscores the continuous efforts to run data-processing tasks faster and quicker in the age of big data. Following are some research or proposed methodologies.

Dheeraj Kumar Singh et al [12] proposed a Graph-based method for extracting frequent patterns of accessing web pages from web server logs, focusing on revealing the complex user browsing scenarios in a graph-by-graph structure to identify underlying patterns. In this method, preprocessing is done on web server logs building a web usage graph followed by graph pruning and finally mining popular sequential access patterns. Pruning techniques applied priorly to data optimization enable us to extract valuable sequential user access patterns from the pruned graph. The future scope of this paper is to extend to personalized advertising, integrating machine learning to increase pattern accuracy and relevance for web usage analysis.

Aatif Jamshed, et.al [1] proposed a hybrid method that combines Convolutional Neural Networks (CNNs) [1] and Long Short-Term Memory (LSTM) [1] networks to extract patterns more effectively and precise than traditional Sequential pattern mining (SPM). The main stages in this method are data preprocessing, non-stationary data conversion by using Discrete Wavelet Analysis, and applying sequential pattern mining with CNN and LSTM. Experimental results has that this proposed method is more effective in solving large-scale sequential pattern mining tasks.

The paper by Meghna Sharma et.al [13] addresses outlier prediction in Radio Frequency Identification (RFID)-enabled supply chain paths. RFID technology is important for Supply Chain Management due to its non-line-of-sight reading features and automatic long-distance sensing capabilities. Conventional models, including HMM and XGBoost, face major problems with longer sequences. As a result, the study seeks the possibility of using LSTM networks to improve the accuracy of the predictions. The proposed architecture comprises pre-processing and feature extraction followed by LSTM-based outlier prediction, which is experimentally proven to perform better than other models. LSTM-based approaches are proposed to improve the efficiency and performance of the supply chain, stressing the need for incorporating modern techniques into modern supply chain management. This research proposes a novel approach to constantly researching and developing innovative technology and techniques for maximizing the efficiency of supply chain operations.

In the realm of early warning for steam turbine stages Xingshuo Li, et.al [14] advocated for sophisticated machine learning techniques above conventional models. In the early stages, the methods were about performance monitoring parameters but they had low adaptability to dynamic conditions and were replaced by artificial neural networks. LSTM networks are increasingly used to characterize time-series changes rather than capture temporal dynamics, improving detection accuracy during transient operations. With their adaptive ability to dynamically consider attributes and extract the time series data, LSTM networks are the preferred choice for modeling the frequent patterns of stage performance of steam turbines. The most critical points are features choosing and fusing, preservation of the model complexity, and information richness. Challenges persist in scalability, interpretability, and generalization. Future research efforts may focus on hybrid modeling approaches and developing frameworks for trend forecasting and remaining useful life estimation, facilitating effective condition-based maintenance practices.

METHODOLOGY

Understanding user behaviour on web platforms is essential in the digital age to optimize system performance and improve user experience. This methodology focuses on using graph-based techniques to analyze web access patterns in order to obtain insights that can inform system design and optimization strategies. Finding frequent patterns in web access sequences is our aim. In addition to providing useful data regarding user behaviour, these patterns and the metrics that correlate with them serve as training data for Long Short-Term Memory (LSTM) models. This model, which offers an effective tool for anticipating and comprehending user behaviour in web applications, is evaluated for predicting frequent patterns after being trained on the extracted patterns. These are the actions taken to put our design into implementation.

1. Data Collection and Preprocessing:

The data set consists of Rack ID, Application ID, User ID, Data Node ID, Block ID, File ID, and timestamp which are organized by racks using DFS architecture. This data is rearranged into separate Data Frame based on Rack Id and Data Node Id combination This organization enables detailed analysis of each rack and data node. Consecutive occurrences of the same Block ID within the same session are removed to ensure data integrity. Now this data is used in graph-based frequent pattern generation.

2. Graph Representation:

The create graph function converts web access sequences into a graph representation that captures the dynamic structure of the web access sequence, which is a key-value pair where the key represents a Session Id. The value is a list of sequences accessed during that session. In graph web pages are represented by nodes, edges represent page transitions, and edge weight is the frequency of page transition

3. Pruning Graph:

The min count function is used to prune the graph, removing nodes and edges with counts less than a specified threshold. After pruning the graph Node Id and edge Id are assigned as unique identifiers to the remaining nodes and edges. The Prune Graph function returns a pruned graph which is optimized.

4. Pattern Mining:

The pattern frequency is identified for individual Rack Id and Data Node Id sets using the Mine Graph function. Patterns illustrate sequences of page views with high repetition rates which are commonly repeated in graphs of the web access. Frequent Patterns are extracted including their frequency, length, and support, providing their significance and frequency within the dataset.

5. Data Preparation:

The Frequent patterns that are extracted from pattern Mining along with their respective Frequency, Length, Support, Rack Id, and Data Node Id are saved in a CSV file for further processing as the training dataset. The data for testing is taken from the collected dataset. The test and training datasets are passed to the LSTM model for frequent pattern prediction.

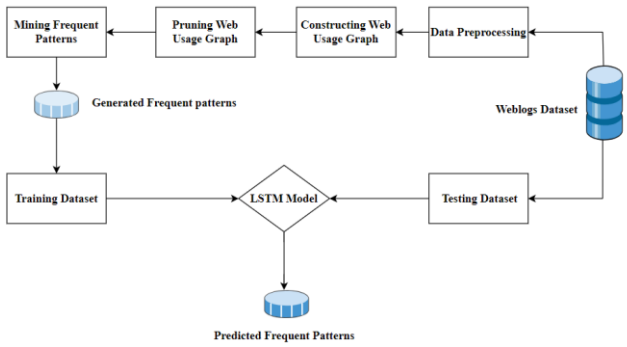

Fig. 2. A flow chart of the proposed Architecture

6. Model Training:

The Long Short-Term Memory (LSTM) model predicts patterns from the data train set. Encoding is performed to train and test the dataset which trains the model to extract patterns and validates from the test dataset which enables it to make accurate predictions.

7. Model Evaluation:

The performances of the LSTM model are assessed by comparing the patterns predicted for training with the patterns in the test data. Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Square Error (MSE) is computed to measure the model's proficiency and effectiveness in predicting Web access patterns.

RESULTS AND DICUSSION

The entailed the development of an algorithm that combined machine learning optimization with a graph-based methodology to efficiently extract Frequent Patterns from data residing within a distributed file system (DFS). In particular, the algorithm represented web access sequences using graph structures, which enabled the identification of recurring patterns and facilitated read-time optimization in DFS

scenarios. Following rigorous testing and evaluation, the algorithm demonstrated a significant decrease in mean read access time, which surpassed the performance of traditional approaches.

Table I. Performance metrics comparison.

Metric	Formula	Value
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$	32.657
MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$	1066.498
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i-\hat{y}_i\rvert$	32.657

The application of graph-based methodologies has been demonstrated to be an effective approach in capturing data relationships and usage trends within Distributed File System (DFS) structures. A notable process that has arisen from this approach is the pruning of infrequent nodes and edges, resulting in the amplification of the significance of generated patterns and the development of more efficient prefetching algorithms. Furthermore, the integration of machine learning models, particularly Long Short-Term Memory (LSTM) networks, has facilitated the accurate forecasting of future patterns, thus optimizing DFS performance further.

These findings demonstrate the potential of the proposed method in improving read-time operations in practical DFS applications. However, future research could focus on exploring more advanced machine learning models and improving scalability to enable wider applicability across various data-intensive domains. By doing, the proposed method could be further developed to enhance the efficiency of DFS structures, and as a result, improve their overall performance.

CONCLUSION

In conclusion, the research demonstrates how a graph-based algorithm for frequent pattern generation, coupled with machine learning optimization techniques, can be used to enhance the read operations within DFS.LSTM model was trained by using the data generated by the graph algorithm, which is composed of key features such as Rack ID, Data Node ID, support, frequency, frequent patterns, and length of the frequent pattern. The training demonstrated the model's capability to predict frequent patterns for test data. These predicted patterns play an important role in optimizing read and write operations in Distributed File Systems, implying the substantial potential for enhancing data management and analysis in the context of big datasets. The conclusion of this study underlines the need of employing sophisticated data processing techniques to deal with the difficulties of processing and analyzing enormous amounts of data. Through the use of graph-based algorithms and machine-learning models effectively, it not only preserved the effectiveness of distributed file systems but also expanded the scope of their application to recommendation systems and data-intensive domains.

FUTURE SCOPE

In the future, some areas will need further improvement and research to achieve better optimization results and faster data transfer operations. In addition to this, the implementation of more complicated machine learning methods, like deep learning architectures, could increase the model's predictive potential to a considerable extent. Also, the applications of reinforcement learning algorithms for the dynamic optimization problem in DFS systems are another exciting research direction for future scalability issues that need to be investigated, and the system should be capable of handling large-scale deployments of DFS for practical implementation. Lastly, the need to come up with privacy-preserving strategies to secure and maintain confidentiality as well as security in DFS needs to be explored further in future research.

REFERENCES

[1] Jamshed, A., Mallick, B. Kumar, P. Deep learning-based sequential pattern mining for progressive database. Soft Comput 24, 17233–17246 (2020). https://doi.org/10.1007/s00500-020-05015-2

[2] Kwon W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., ... & Stoica, I. (2023, October). Efficient memory management for large language model serving with paged attention. In Proceedings of the 29th Symposium on Operating Systems Principles (pp. 611-626).

[3] J.Wong M. H., Tseng, V. S., Tseng, J. C., Liu, S. W., & Tsai, C. H. (2017). Long-term user location prediction using deep learning and periodic pattern mining. In Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings 13 (pp. 582-594). Springer International Publishing.

[4] Li, F., Gui, Z., Zhang, Z., Peng, D., Tian, S., Yuan, K., ... Lei, Y. (2020). A hierarchical temporal attention-based LSTM encoder-decoder model for individual mobility prediction. Neurocomputing, 403, 153-166.

[5] Moldovan,D., Anghel, I., Cioara, T. , Salomie, I., (2019, October) Time series features extraction versus lstm for manufacturing processes per- formance prediction. In 2019 international conference on speech technology and human-computer dialogue (SpeD) (pp. 1-10). IEEE.

[6] Hajiaghayi, M., Vahedi, E., (2019, April) Code failure prediction and pattern extraction using LSTM networks. In 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataSer- vice)(pp. 55-62). IEEE.

[7] Padmavathi, B., Bhagyalakshmi, A., Kavitha, D., Indumathy, P., (2024) An optimized Bi-LSTM with random synthetic over-sampling strategy for network intrusion detection. Soft Computing, 28(1), 777- 790.

[8] Houetohossou, S. C. A., Ratheil Houndji, V., Sikirou, R., Gl `el `e Kaka¨ı, R., (2024) Finding optimum climatic parameters for high tomato yield in Benin (West Africa) using frequent pattern growth algorithm. Plos one, 19(2), e0297983.

[9] Vijayalakshmi, S., Mohan, V., Raja, S. S., (2010) Mining of users access behavior for frequent sequential pattern from web logs. International Journal of Database Management System (IJDM), 2.

[10] Singh, D. K., Sharma, V., Sharma, S., (2012) Graph based approach for mining frequent sequential access patterns of web pages. International Journal of Computer Applications, 40(10), 33-37.
[11] Sharma, M., Tomer, M. S., (2018) Predictive analysis of RFID supply chain path using long short term memory (LSTM): recurrent neural networks. International Journal of Wireless and Microwave Technologies (IJWMT), 8(4), 66-77.
[12] Li, X., Liu, J., Bai, M., Li, J., Li, X., Yan, P., Yu, D., (2021) An LSTM based method for stage performance degradation early warning with consideration of time-series information. Energy, 226, 120398.
[13] Jahani, A., Zare, K., & Khanli, L. M. (2023). Short-term load forecasting for microgrid energy management system using hybrid SPM-LSTM. *Sustainable Cities and Society*, *98*, 104775
[14] Subramanian, M., & Rajalakshmi, V. R. (2023, July). Deep Learning Approaches for Melody Generation: An Evaluation Using LSTM, BILSTM and GRU Models. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE..
[15] Kim, J., & Moon, N. (2020). LSTM-Based Consumption Type Prediction Model. In *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 2018* (pp. 564-567). Springer Singapore.
[16] Wang, X., & Kadioglu, S. (2022). Dichotomic pattern mining with applications to intent prediction from semi-structured clickstream datasets. *arXiv preprint arXiv:2201.09178*.
[17] Xia, B., Bai, Y., Yin, J., Li, Y., & Xu, J. (2021). Loggan: a log-level generative adversarial network for anomaly detection using permutation event modeling. *Information Systems Frontiers*, *23*, 285-298.
[18] Ibrahim, Z. M., Bean, D., Searle, T., Qian, L., Wu, H., Shek, A., ... & Dobson, R. J. (2021). A knowledge distillation ensemble framework for predicting short-and long-term hospitalization outcomes from electronic health records data. *IEEE Journal of Biomedical and Health Informatics*, *26*(1), 423-435.
[19] Tax, N. (2018, June). Human activity prediction in smart home environments with LSTM neural networks. In *2018 14th international conference on intelligent environments (IE)* (pp. 40-47). IEEE.
[20] Crivellari, A., & Beinat, E. (2020). LSTM-based deep learning model for predicting individual mobility traces of short-term foreign tourists. *Sustainability*, *12*(1), 349.