Erasure Coding Technique for Data Replication in HDFS

P. Haritha[1] Mrs. C. Hemavathy[2]
[1]Student [2]Assistant Professor
[1,2,3,4,5]Department of Computer Applications
[1,2,3,4,5]KMM Institute of PG Studies, Tirupati, India

Abstract— The Hadoop Distributed File System (HDFS) component of Apache Hadoop helps in distributed storage of big data with a cluster of commodity hardware. HDFS ensures availability of data by replicating data to different nodes. However, the replication policy of HDFS does not consider the popularity of data. The popularity of the files tends to change over time. Hence, maintaining a fixed replication factor will affect the storage efficiency of HDFS. In this paper we propose an efficient dynamic data replication management system, which consider the popularity of files stored in HDFS before replication. This strategy dynamically classifies the files to hot data or cold data based on its popularity and increases the replica of hot data by applying erasure coding for cold data. The experiment results show that the proposed method effectively reduces the storage utilization up to 40% without affecting the availability and fault tolerance in HDFS.

Key words: Big Data, Hadoop Distributed File System, Dynamic Data Replication

I. Introduction

Big Data is high-volume, high-velocity and high-variety information that demands cost-effective, innovative forms of information processing for enhanced insight and decision making. The extraction, storage and processing of big data is beyond the ability of traditional data processing techniques. Therefore, a more sophisticated framework is required to handle these data. Apache Hadoop is one of the best-known platforms for distributed storing and processing of big data across clusters of computers. The storage component of Hadoop, Hadoop Distributed File System (HDFS) maintains a default replication factor for each file as three, which is placed in separate nodes. HDFS provides high performance access to data by applying a static and default replication strategy. Though HDFS ensures high reliability, scalability and high availability, it's static and default approach in data replication requires large amount of storage space. With a replication factor of three, a file is copied three times in different nodes. If the size of a file is 1TB then, after replication it will take 3TB of space. Furthermore, in HDFS the files are replicated without considering the popularity of the file. In real scenario, the access frequency of every file in the file system is not accessed equally. Some files are accessed frequently while some others stay idle for a long period of time. In the proposed work, we present a dynamic data replication strategy which focuses on a storage efficient replication in HDFS without affecting the availability of data. In this strategy data files are classified into hot and cold based on the popularity of the data file in the Hadoop cluster. The replica of the popular file is increased while the replica of non-popular file is reduced to one and erasure coding is applied on it to prevent from data loss. The result shows that the proposed replication strategy reduces the storage space utilization significantly without affecting the availability constraint of HDFS. The remainder of this paper is structured as follows. In section 2, we discuss the related work and in section 3, we discuss the background theory. In section 4, we present the data replication strategy in detail with its architecture and algorithm. In section 5, we include the implementation and evaluation results demonstrating the storage efficiency of our proposed algorithm. Finally, in section 6, we conclude with the scope of future work. By keeping replicas for these idle files, a valuable amount of storage space will be consumed. Unnecessarily resulting in wastage of storage space that leads to bad effect of performance. If the number of copies can be reduced for these files, the storage space can be freed and can be utilized by more frequently accessed files. But reducing the number of copies increases the chance for data loss. Therefore, a data replication strategy which reduces the replicas of under-utilized file without affecting the data availability and fault tolerance has to be implemented.

II. Literature Survey

A. Adaptive Replication Management in HDFS based on Supervised Learning:

The number of applications based on Apache Hadoop is dramatically increasing due to the robustness and dynamic features of this system. At the heart of Apache Hadoop, the Hadoop Distributed File System (HDFS) provides the reliability and high availability for computation by applying a static replication by default. However, because of the characteristics of parallel operations on the application layer, the access rate for each data file in HDFS is completely different. Consequently, maintaining the same replication mechanism for every data file lead to detrimental effects on the performance. By rigorously considering the drawbacks of the HDF replication this paper proposes an approach to dynamically replicate the data file based on the predictive analysis. With the help of probability theory, the utilization of each data file can be predicted to create a corresponding replication strategy. Eventually, the popular files can be subsequently replicated according to their own access potentials. For the remaining low potential files, an erasure code is applied to maintain the reliability. Hence, our approach simultaneously improves the availability while keeping the reliability in comparison to the default scheme. Furthermore, the complexity reduction is applied to enhance the effectiveness of the prediction when dealing with Big Data.

B. Enabling proactive data management in virtualized hadoop clusters based on predicted data activity patterns:

Hadoop clusters are gaining more and more in popularity based on their ability to parallelize and complete large-scale computational tasks on big data. Service offerings of this type have appeared in the recent years, covering a need for dynamic and on-demand creation of such data analytics frameworks. The aim of this paper is to provide a mechanism for offering such virtual clusters as a service, with built-in intelligence functionalities for efficient management. The target of these mechanisms is to predict future demand of the files in the HDFS cluster and dynamically manipulate the according replication factor for availability purposes, in order to improve performance and minimize storage overhead. To this end, real data have been utilized as a dataset input to the prediction framework, based on Fourier series analysis, due to the latter's ability to capture different periodicities that can influence service usage. Multiple time-step ahead prediction is performed in order to enable proactive management (e.g. suitable replication strategy). We describe the framework's architecture, necessary modifications to the client side of Apache Hadoop for data logging and the results of the applied method on two real world datasets.

C. An elastic replication management system for HDFS. In Cluster Computing Workshop:

The Hadoop Distributed File System (HDFS) is a distributed storage system that stores large-scale data sets reliably and streams those data sets to applications at high bandwidth. HDFS provides high performance, reliability and availability by replicating data, typically three copies of every data. The data in HDFS changes in popularity over time. To get better performance and higher disk utilization, the replication policy of HDFS should be elastic and adapt to data popularity. In this paper, we describe ERMS, an elastic replication management system for HDFS. ERMS provides an active/standby storage model for HDFS. It utilizes a complex event processing engine to distinguish real-time data types, and then dynamically increases extra replicas for hot data, cleans up these extra replicas when the data cool down, and uses erasure codes for cold data. ERMS also introduces a replica placement strategy for the extra replicas of hot data and erasure coding parities. The experiments show that ERMS effectively improves the reliability and performance of HDFS and reduce storage overhead.

III. PROPOSED MODEL

The proposed algorithm was measured for each time interval by using the same test data. The graphical representation of the storage space utilized while experimenting with the Hadoop's default three replication and proposed replication algorithm using the sample data set over a specified time interval.

A. Application Architecture:

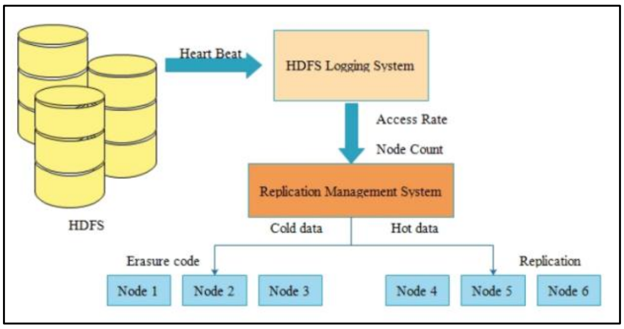
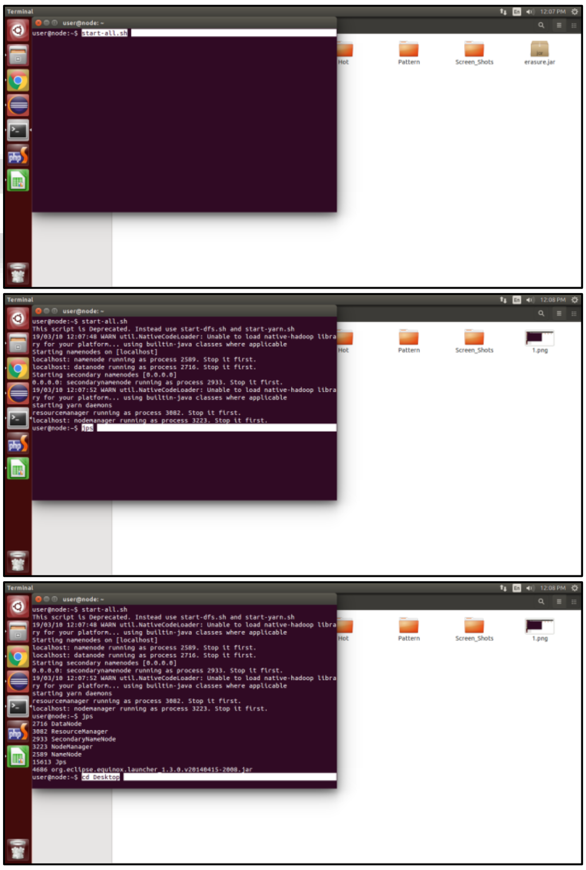


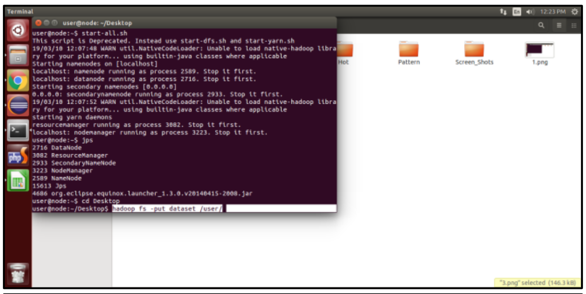
Fig. 1: Architecture of the Proposed System

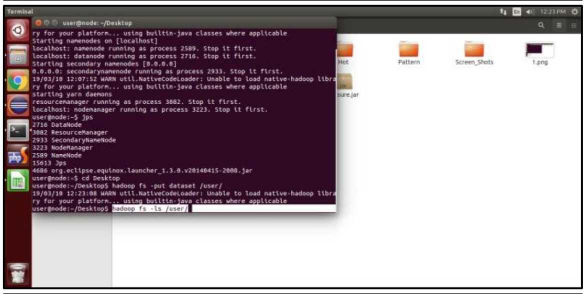
IV. PROPOSED ALGORITHM:

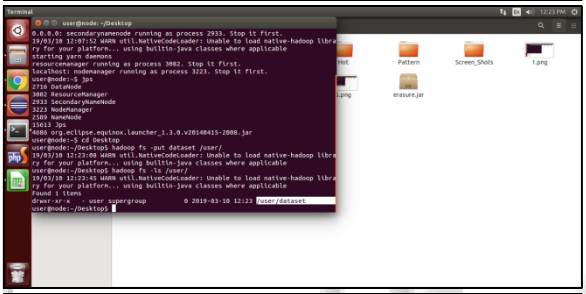
Changing Data Replication Algorithm
Input: log Begin
1) Set time interval
2) For each time interval
{
   1) read logfile
   2) for each file fi
      {
      a) Find aci, nci, rfi
      b) Calculate popularity index (PIi) of each file
      PIi = (aci * nci) / rfi
      } iii. Calculate the threshold,
   3) For each file fi
   4) Compare threshold T If PIi >= T hd fi Else cd fi v.
   For each fi in hd,
   Increment rfi by 1. vi.
For each fi in cd
Set rfi to1
Encode fi using Reed-Solomon erasure code
} end for
End

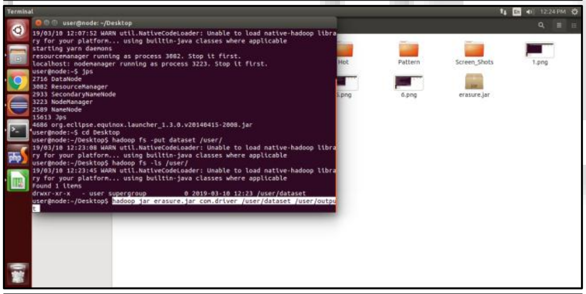
V. RESULTS & ANALYSIS

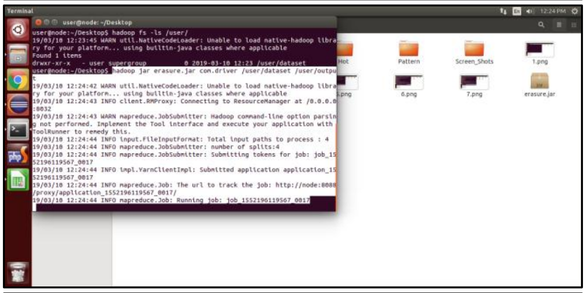


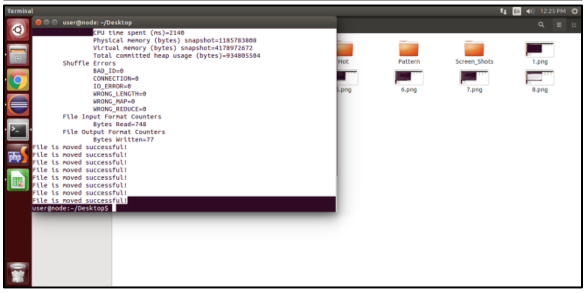


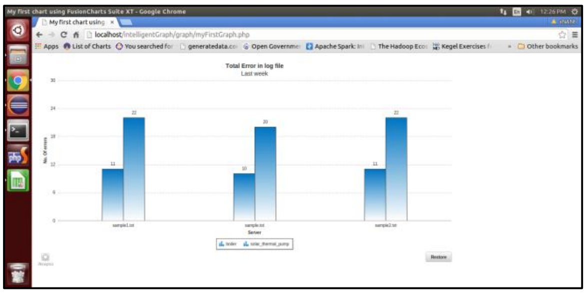












VI. CONCLUSION

HDFS is equipped with a mechanism that uniformly replicates every file without considering the popularity of the file. However, this replication strategy still remains a critical drawback with regards to the storage aspect. To overcome this drawback a storage efficient dynamic data replication strategy is implemented which can dynamically adapt to changes in data popularity. In this work, the storage space is optimized by reducing replication factor of the cold files and applying erasure code. Application of this strategy results in substantial storage-cost savings in hardware expenditure. This work can be improved further by optimizing the placement strategy of replicas in Hadoop cluster. Also, a comparison can be made based on the performance of proposed strategy and existing method in Hadoop.

REFERENCE

[1] The Gartner website [Online]. Available: http://www.gartner.com/it-glossary/big-data/.

[2] The Apache Hadoop website. [Online]. http://hadoop.apache.org/.

[3] The Apache Hadoop Guide website. [Online]. http://hadoop.apache.org/common/docs/stable/hdfs_design.html.

[4] Shvachko, K., Kaung, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on (pp. 1-10). IEEE.

[5] Wei, Q., Veeravalli, B., Gong, B., Zeng, L., & Feng, D. (2010, September). CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster. In Cluster Computing (CLUSTER), 2010 IEEE International Conference on (pp. 188-196). IEEE.

[6] Ananthanarayanan, G., Agarwal, S., Kandula, S., Greenberg, A., Stoica, I., Harlan, D., & Harris, E. (2011, April). Scarlett: coping with skewed content popularity in map reduce clusters. In Proceedings of the sixth conference on Computer systems (pp. 287-300). ACM

[7] Abad, C. L., Lu, Y., & Campbell, R. H. (2011, September). DARE: Adaptive data replication for efficient cluster scheduling. In Cluster Computing (CLUSTER), 2011 IEEE International Conference on (pp. 159-168). IEEE

[8] Kaushik, R. T., Abdelzaher, T., Egashira, R., & Nahrstedt, K. (2011, July). Predictive data and energy management in GreenHDFS. In Green Computing Conference and Workshops (IGCC), 2011 International (pp. 1-9). IEEE.

[9] "Bsoul, M., Al-Khasawneh, A., Abdallah, E. E., & Kilani, Y. (2011). Enhanced fast spread replication strategy for data grid. Journal of Network and Computer Applications, 34(2), 575- 580.
[10] Cheng, Z., Luan, Z., Meng, Y., Xu, Y., Qian, D., Roy, A., & Guan, G. (2012, September). Erms: An elastic replication management system for hdfs. In Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on (pp. 32-40). IEEE.
[11] Kousiouris, G., Vafiadis, G., & Varvarigou, T. (2013, October). Enabling proactive data management in virtualized hadoop clusters based on predicted data activity patterns. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on (pp. 1-8). IEEE.
[12] Papoulis, A. (1977). Signal analysis (Vol. 191). New York: McGraw-Hill.
[13] Bui, D. M., Hussain, S., Huh, E. N., & Lee, S. (2016). Adaptive Replication Management in HDFS based on Supervised Learning. IEEE Transactions on Knowledge and Data Engineering, 28(6), 1369-1382.
[14] Qu, K., Meng, L., & Yang, Y. (2016, August). A dynamic replica strategy based on Markov model for hadoop distributed file system (HDFS). In Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on (pp. 337-342). IEEE.
[15] Reed, Irving S.; Solomon, Gustavo (1960), Polynomial Codes over Certain Finite Fields, Journal of the Society for Industrial and Applied Mathematics (SIAM), 8 (2): 300–304, doi:10.1137/0108018