Available online at: http://www.ijmtst.com/vol3issue10.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 03, Issue No: 10, October 2017

Dynamic Data Deduuplication and Replication with HDFS: using Big Data

D.Leela Dharani[1] | Harika Pothini[2] | B.Srinivas[3]

[1]Assistant Professor, Department of Information Technology, PVP Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India.

[2]Lecture, Department of Computer Science, Sri Durga Malleswara Siddartha Mahila Kalasala, Vijayawada, Andhra Pradesh, India.

[3]Assistant Professor, Department of MCA, VR Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India.

To Cite this Article

D.Leela Dharani, Harika Pothini and B.Srinivas, "Dynamic Data Deduuplication and Replication with HDFS: using Big Data Flow Test to Analyze Fuel Filter Element of an Aircraft", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 10, October 2017, pp: 133-137.

ABSTRACT

Now-a-days Apache Hadoop could be very famous applications based totally on Apache Hadoop are growing in nowadays due to its lively and different functions. Hadoop dispensed file gadget(HDFS) is the heart of Apache Hadoop which is reliable and exceptionally to be had.it's far having static Replication approach by way of default.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, its 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. 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 50% without affecting the availability and fault tolerance in HDFS. Because of this dynamic technique, specific Replication approach and Erasure Code Mechanism to improve Availability and Reliability.

***Keywords** — Apache Hadoop, Dynamic data replication, Optimization, reliability.*

I. INTRODUCTION

An adaptive replication management (ARM) system is designed to provide high availability for the data in HDFS via enhancing the data locality metric. As a result, the highly local available data improves the performance of the Hadoop system. It is worth noting that the erasure code is applied to maintain the reliability. A complexity reduction method for the prediction technique is proposed in both hyper-parameter learning and training phases. This proposed method significantly increases the performance in terms of reaction rate for the replication strategy while still keeping the accuracy of the prediction.ARM in HDFS is implemented here and an evaluation is done in order to practically verify the effectiveness of the proposed method as compared with the state of the art method. Big data has become the new frontier of information management given the amount of data today's systems are generating and consuming. It has driven the need for technological infrastructure and tools that can capture, store, analyse and visualize vast amounts of disparate structured and unstructured data. These data are being generated at increasing volumes from data intensive technologies including, but not limited to, the use of the Internet for activities such as accesses to information, social networking, mobile computing and commerce. Corporations and governments have begun to recognize that there are unexploited opportunities to improve their enterprises that can be discovered from these data. Most corporate enterprises face significant challenges in fully leveraging their data. Frequently, data is locked away in multiple databases and processing systems throughout the enterprise, and the questions customers and analysts ask require an aggregate view of all data, sometimes totalling hundreds of terabytes.

II. LITERATURE SURVEY

Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng

Data replication has been widely used as a mean of increasing the data availability of large-scale cloud storage systems where failures are normal. Aiming to provide cost-effective availability, and improve performance and load-balancing of cloud storage, this paper presents a cost-effective dynamic replication management scheme referred to as CDRM. A novel model is proposed to capture the relationship between availability and replica number. CDRM leverages this model to calculate and maintain minimal replica number for a given availability requirement. Replica placement is based on capacity and blocking probability of data nodes. By adjusting replica number and location according to workload changing and node capacity, CDRM can dynamically redistribute workloads among data nodes in the heterogeneous cloud. We implemented CDRM in Hadoop Distributed File System (HDFS) and experiment results conclusively demonstrate that our CDRM is cost effective and outperforms default replication management of HDFS in terms of performance and load balancing for large-scale cloud storage.

G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris

To improve data availability and resilience MapReduce frameworks use file systems that replicate data *uniformly*. However, analysis of job logs from a large production cluster shows wide disparity in data popularity. Machines and racks storing popular content become bottlenecks; thereby increasing the completion times of jobs accessing this data even when there are machines with spare cycles in the cluster. To address this problem, we present Scarlett, a system that replicates blocks based on their popularity. By accurately predicting file popularity and working within hard bounds on additional storage, Scarlett causes minimal interference to running jobs. Trace driven simulations and experiments in two popular MapReduce frameworks (Hadoop, Dryad) show that Scarlett effectively alleviates hotspots and can speed up jobs by 20.2%.

K. S. Esmaili, L. Pamies-Juarez, and A. Datta

Erasure codes are an integral part of many distributed storage systems aimed at Big Data, since they provide high fault-tolerance for low overheads. However, traditional erasure codes are inefficient on reading stored data in degraded environments (when nodes might be unavailable), and on replenishing lost data (vital for long term resilience). Consequently, novel codes optimized to cope with distributed storage system nuances are vigorously being researched. In this paper, we take an engineering alternative, exploring the use of simple and mature techniques -juxtaposing a standard erasure code with RAID-4 like parity. We carry out an analytical study to determine the efficacy of this approach over traditional as well as some novel codes. We build upon this study to design CORE, a general storage primitive that we integrate into HDFS. We benchmark this implementation in a proprietary cluster and in EC2. Our experiments show that compared to traditional erasure codes, CORE uses 50% less bandwidth and is up to 75% faster while recovering a single failed node, while the gains are respectively 15% and 60% for double node failures.

Cerri et al proposed 'Knowledge in the cloud' in place of 'data in the cloud' to support collaborative tasks which are computationally intensive and facilitate distributed, heterogeneous knowledge. This is termed as "Utility Computing" derived from required data in and out of Cloud the utilities like electricity, gas for which we only pay for what we use from a shared resource. With the growing interest in cloud, analytics is a challenging task.In general, Business Intelligence applications such as image processing, web searches, understanding customers and their buying habits, supply chains and ranking and Bio-informatics (e.g. gene structure prediction) are data intensive applications. Cloud can be a perfect match for handling such analytical services. For example, Google's MapReduce can be leveraged for analytics as it intelligently chunks the data into smaller storage units and distributes the computation among low-cost processing units. Several research teams have started working on creating Analytic frameworks and engines which help them provide Analytics as a Service. For example, Zementis launched the ADAPA predictive analytics decision engine on Amazon EC2, allowing its users to deploy, integrate, and execute statistical scoring models like neural networks, support vector machine (SVM), decision tree, and various regression models.

III. PROPOSED SYSTEM

Theoretically, by placing the potential replicas on low utilization nodes (low blocking rate nodes), the replication management helps to redirect the tasks to these idle nodes and balance the computation. The blocking rate is calculated based on the information provided by the monitoring system. Based on Ganglia framework , the monitoring system is simple, robust and easy to configure for monitoring most of the required metrics. After plugging into the HDFS nodes, the monitoring system can collect statistics via Ganglia API.

- We designed an adaptive replication management (ARM) system to provide high availability for the data in HDFS via enhancing the data locality metric. As a result, the highly local available data improves the performance of the Hadoop system. It is worth noting that the erasure code is applied to maintain the reliability.
- We proposed a complexity reduction method for the prediction technique in both hyper-parameter learning and training phases. This proposed method significantly increases the performance in terms of reaction rate for the replication strategy while still keeping the accuracy of the prediction.
- We implemented ARM in HDFS and did an evaluation in order to practically verify the effectiveness of the proposed method as compared with the state of the art method.

ADVANTAGES OF PROPOSED SYSTEM:

- The main function of the proposed architecture is to dynamically scale the replication factors as well as to efficiently schedule the placement of replicas based on the access potential of each data file.
- Additionally, to reduce the calculation time, the knowledge base and heuristic technique are implemented to detect the similarity in the access pattern between in-processing files and the predicted ones.
- By definition, the access pattern is actually a set of eigenvectors describing the feature properties of processed data.
- Two files with similar access behaviors are treated with the same replication strategy. However, because these techniques are minor parts and popularly used in various systems, discussing them is not within the scope of this paper.

IV. REALATED WORK

In HDFS, in order to ensure data availability and to reduce the chance of data loss, each file is replicated across a number of machines. The default replication factor in HDFS is to create three replicas for each file. HDFS replication strategy will not consider whether a particular file is popular or not. Unnecessary replication of non-popular file will result in storage overhead. In the proposed strategy, a dynamic data replication algorithm is used to manage the replicas in HDFS. The Replication Management System in the proposed algorithm manages the replication of files in HDFS. This module classifies the data files into hot data or cold data. After classifying the data, the replication factor is increased for the hot data and its replica is placed in data node. For placement of replicated data Hadoop's random placement strategy is used. Erasure coding is applied for cold data to ensure availability. Replication Management System does these tasks with the help of HDFS Logging System. The logging system provide details such as the number of files accessed, their source, the nodes which accessed them, frequency of access for each file, etc. The Logging system obtains all these information from HDFS and provides it to the Replication management System. a Hadoop cluster was setup comprising of 10 nodes. The physical Hadoop cluster comprises of one master node and nine slave nodes and the version of a Hadoop distribution is 2.7. The master node acts as both name node and data node, thus a cluster of ten data nodes is formed. Each node is equipped with Intel
Core i5 (3.30GHz) CPU and 8 GB RAM. Files were copied into HDFS from the local file system. Files of different size and types are considered for the

experimentation of the algorithm. Various types of files including text, audio, video files with sizes ranging from The 600MB to 4GB were considered for this purpose. The files were divided into blocks by HDFS with default block size of 128MB. Initially, the replications for the files were three, which is the default replication count in HDFS. These files were accessed randomly from different nodes, at different time intervals. The log files were analyzed. Then in regular intervals the algorithm is executed in the HDFS cluster. The algorithm checks the access count for each file and calculates their popularity. Based on this popularity value and threshold, files are classified into two – hot or cold. Replication count for hot files is incremented and cold files are encoded using Reed-Solomon erasure code. The performance of the algorithm was analy zed comparing the result obtained by using Hadoop default replication strategy .

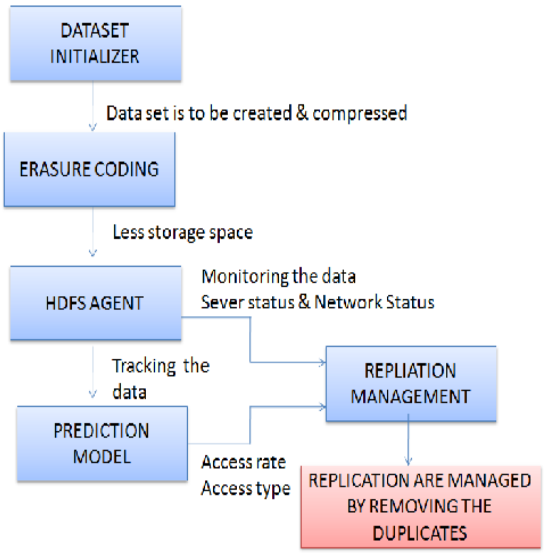

Fig:-Block Diagram of HDFS

Proposed Dynamic Data Replication Algorithm:

Input: log
Begin
1. Set time interval
2. For each time interval
{
i. read logfile
ii. 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,
iv. For each file fi
Compare threshold T
If PIi >= T fi⇓
hd Else fi
⇓cd
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. CONCLUSION

In this paper, to improve the availability of HDFS by enhancing the data locality, our contribution focuses on following points. First, we design the replication management system which is truly adaptive to the characteristic of the data access pattern. The approach not only pro-actively performs the replication in the predictive manner, but also maintains the reliability by applying the erasure coding approach. Second, we propose a complexity reduction method to solve the performance issue of the prediction technique. In fact, this complexity reduction method significantly accelerates the prediction process of the access potential estimation. Finally, we implement our method on a real cluster and verify the effectiveness of the proposed approach. With a rigorous analysis on the characteristics of the file operations in HDFS, our uniqueness is to create an adaptive solution to advance the Hadoop system.

REFERENCES

[1] "What is apache hadoop?" https://hadoop.apache.org/, accessed: 2015-08-13.

[2] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," in Proceedings of the 5th European conference on Computer systems. ACM, 2010, pp. 265–278.

[3] K. S. Esmaili, L. Pamies-Juarez, and A. Datta, "The core storage primitive: Cross-object redundancy for efficient data repair & access in erasure coded storage," arXiv preprint arXiv:1302.5192, 2013.

[4] G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris, "Scarlett: Coping with skewed content popularity in mapreduce clusters." in Proceedings of the Sixth Conference on Computer Systems, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 287–300.

[Online]. Available: http://doi.acm.org/10.1145/1966445.1966472

[5] G. Kousiouris, G. Vafiadis, and T. Varvarigou, "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, Oct 2013, pp. 1–8.

[6] A. Papoulis, Signal analysis. McGraw-Hill, 1977, vol. 191.

[7] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "Cdrm: A cost-effective dynamic replication management scheme for cloud storage cluster." in Cluster Computing (CLUSTER), 2010 IEEE International Conference on, Sept 2010, pp. 188–196.

[8] C. L. Abad, Y. Lu, and R. H. Campbell, "Dare: Adaptive data replication for efficient cluster scheduling." in CLUSTER. IEEE, 2011, pp. 159–168.

[9] Z. Cheng, Z. Luan, Y. Meng, Y. Xu, D. Qian, A. Roy, N. Zhang, and G. Guan, "Erms: An elastic replication management system for hdfs." in Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, Sept 2012, pp. 32–40.

[10] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: Novel erasure codes for big data," in Proceedings of the VLDB Endowment, vol. 6, no. 5. VLDB Endowment, 2013, pp. 325–336.

[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.