International Journal of Recent Engineering Research and Development (IJRERD)
Volume No. 02 – Issue No. 03, ISSN: 2455-8761
www.ijrerd.com, PP. 56-60

Adaptive Replication Management in HDFS based on Supervised Learning

[1]Prof. Pawar Abhijeet H, [2]Subhedar Dipti D., [3]Phadtare Jigish K., [4]Pawase Amol Ashok, [5]Honrao Sayali D.

Department of Info. Tech.
SVPM's College of Engineering,
Malegaon(Bk)
Baramati, Pune -413104

Abstract: Apache Hadoop is very popular now-a-days.The Applications based on ApacheHadoopare increasing in these days due to its vigorous and dierent features.Hadoop Distributed File System(HDFS) is the Heart of Apache Hadoop which is reliable and highly available.It is having static Replication Strategy by default.In Application Layer,there are parallel operations were done on the le due to which Access Rate for each Data File becomes dierent.This static Replication Strategy leads to Fatal errors on the Performance of the System.To overcome the drawback of HDFS,This paper explains the dynamic approach to Replicate Data Files based on Supervised Learning.Using Probability Distribution,each data le is used to create corresponding Replication Strategy.By measuring Access Potential,High potential Files were going for Replication and remaining Low potential les are entered in thesection of ErasureCode.Due to this dynamic approach,this paper suggests Unique Replication Strategy and Erasure Code Mechanism to improve Availability and Reliability.

General Terms:
1] HDFS(Hadoop Distributed File System)
2] Replication
3] Proactive Prediction
4] Optimization
5] Bayesian Learning
6] Gaussian Process
Keywords: ARM, ERMS, CDRM, OPTIMIS, DARE, HDFS-RAID

1. INTRODUCTION

The system is known as ARM i.e. Adaptive Replication Management System in which mainly concentration is on HDFS (Haddop Distributed File System) and its mechanism. There are two main components of HDFS system as:
1] Replication management
2] File Distribution
The paper worked on the system components and suggests three main approaches:
1] Adaptive Replication Management
2] Erasure Code
3] Distribution of Files

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.

2. PROPOSED ARCHITECTURE

2.1 System Description

The system starts by periodically collecting the heartbeat. After that, this heartbeat is sent to the heuristic detector as the training data. This training data is compared with the access patterns, which are extracted from the predictor component and stored at the knowledge base. If there is a match, the access potential is then retrieved from the pattern and directly passed to the predictor component without any computation. Otherwise, the training data is continuously sent. In that case, most of the computation belongs to the hyper-parameter learning and training phases of the prediction. To solve this issue, the hyper generator is constructed to reduce the computational complexity of the hyper-parameter learning phase. After that, the training phase can start to estimate the access potential. Finally, the access potential of the target file is passed on to the replication management component. In addition, a new pattern is also extracted and stored at the knowledge base for the next evaluation.

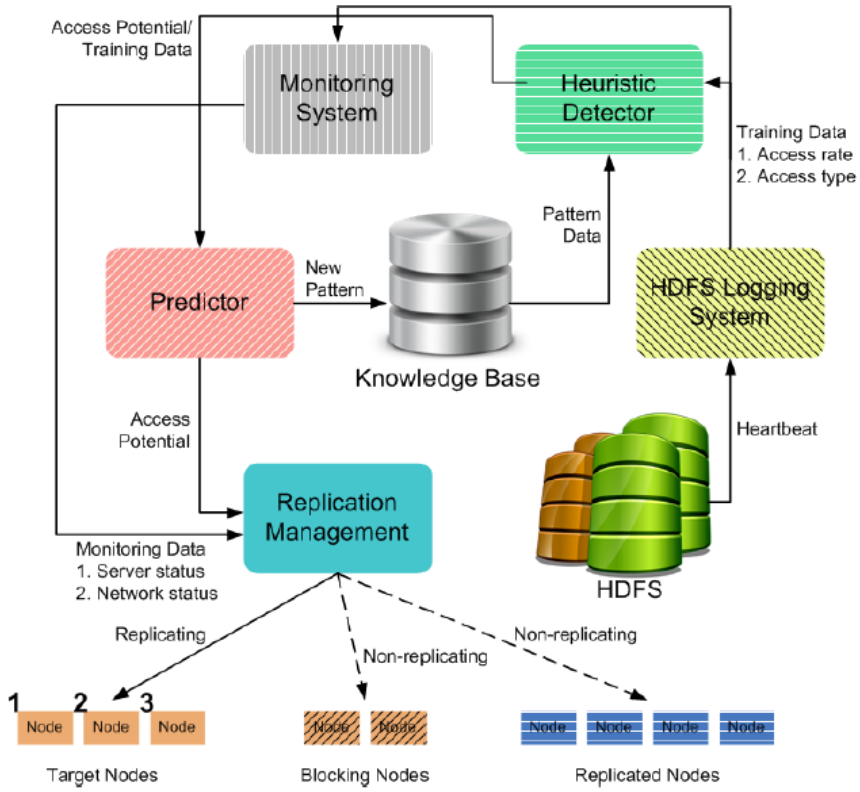

Fig. 1: Architecture of Adaptive Replication Management (ARM) system.

2.2 Replication Management

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.

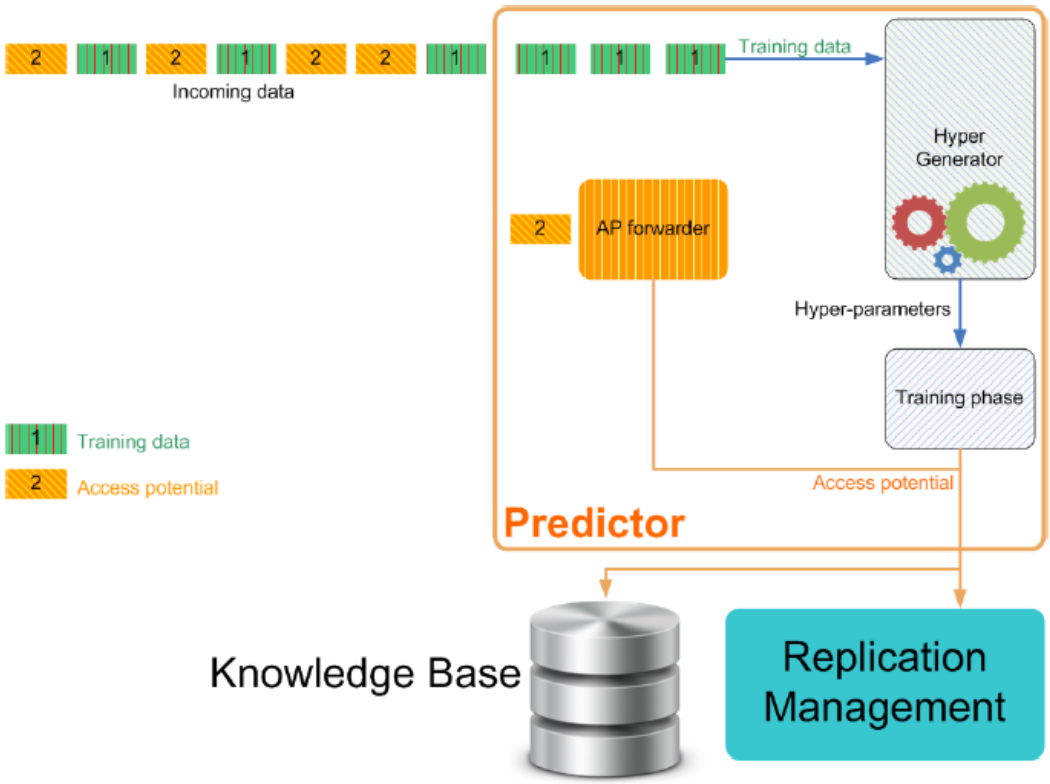

Fig. 2: Working mechanism of Predictor component.

3. WORKING

3.1. Preliminary File Upload :

User first Login to the system with its unique ID and password. Then files were uploaded by the user which are uploaded ,analyzed and
Saved in system.

3.2. Monitoring System :

In this section, scanned files are replicated using the Replication Strategy by predictive analysis. By using the Mechanism, files are divided into two categories :
High Potential Files and Low Potential Files. High Potential Files are going for Replication management and Low Potential les are going for Erasure Code.

3.3. Replication Management:

The popular less can be subsequently replicated according to their own access potentials. Files which were having high potential Rate, are going for the Replication. According to the Access Rate, Replicas were created and going for further Distribution. Distribution in Cluster System .Replicas and the original les are going for Distribution for security reasons. Replicas and the les were stored in the nodes of cluster. They are in non-readable format.

3.4 Erasure Code Mechanism :

Thereafter at last Erasure code mechanism is applied to Low Potential Files. these files are having low access rate therefore already created replicas of this file are erased from the system to improve the Performance of the System.

4. Application:

1. Facebook
2. Windows Azure
3. Yahoo

5. Algorithm

Algorithm 1: Hyper-parameter learning phase

Data: Access array. This is the latest history of access rate of each data file with regards to time step.

Result: Hyper-parameters array $\theta^{(*)} = [l^{(*)}, \sigma_f^{(*)}]$

Initialize value for $\theta^{(0)} = [l^0, \sigma_f^0], \omega, \epsilon_{RMSE}$;

/* Fast Fourier Transform of input data */

$\hat{y}$= nufft1d1(y);

for *k=1* **to** *sizeof($\hat{y}$)* **do**

  /* step_size is equivalent to α in the Equation (22) and (23) */

  step_size=decay_function(k);

  j=random(1,*sizeof*($\hat{y}$));

  /* partial derivative of $\mathcal{F}_{rMLL}$ w.r.t l */

  $\nabla l = \text{partial\_l}(\hat{y}_{[j]}, \omega_{[j]}, l^{(k-1)}, \sigma_f^{(k-1)})$;

  /* partial derivative of $\mathcal{F}_{rMLL}$ w.r.t σ_f */

  $\nabla \sigma_f = \text{partial\_}\sigma_f(\hat{y}_{[j]}, \omega_{[j]}, l^{(k-1)}, \sigma_f^{(k-1)})$;

  /* update hyper-parameters */

  $l^{(k)} = l^{(k-1)}$+step_size*∇l;

  $\sigma_f^{(k)} = \sigma_f^{(k-1)}$+step_size*$\nabla \sigma_f$;

  Compute $\mathcal{F}_{rMLL}^{(k)}(\theta^{(k)})$;

  Compute $RMSE^{(k)} = RMSE(\mathcal{F}_{rMLL}^{(k)})$;

  if $(RMSE^{(k)} \leq \epsilon_{RMSE})$ **then**

    break();

  end

end

return $\theta^{(*)} = [l^{(*)}, \sigma_f^{(*)}]$;

Conclusion:

Following points focuses on the contribution of paper to improve the availability of HDFS to enhance the data locality. The design of replication management system is made firstly which is very much adaptive with the data access pattern characteristics. It maintains the reliability of system using erasure code approach with pro-active performance of the replication in predictive fashion. Performance issue of the prediction technique is solved using a complexity reduction method. This method accelerates the prediction process of access potential estimation. The execution method on real cluster verifies the effectiveness of the proposed approach. The adaptive solution for Hadoop system is the key of a paper with abrasive analysis on the characteristics of the file operations in HDFS.

Acknowledgments:

The work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIP) NRF-2014R1A2A2A01003914 and by the MSIP (Ministry of Science, ICT & Future

Planning), Korea under the ITRC (Information Technology Research Center) support program (NIPA-2014(H0301-14-1020) supervised by the NIPA (National IT Industry Promotion Agency).

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 5thEuropean conference on Computer systems
[3]. K. S. Esmaili, L. Pamies-Juarez, and A. Datta, "The core storageprimitive: 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 theSixth 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 proactivedata management in virtualized hadoop clusters based on predicted data activity patterns." in P2P, Parallel, Grid, Cloud andInternet Computing (3PGCIC), 2013 Eighth International conferenceon, 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: Acost-effective dynamic replication management scheme for cloud storage cluster." in Cluster Computing (CLUSTER), 2010 IEEE InternationalConference on, Sept 2010, pp. 188–196.