ISSN 2395-1621

Flexible Replication Management In Hadoop Database File System

Ms. Surwase Vaishnavi, Ms. Kolpe Arati, Ms. Nawale Anjali, Ms. Anita Thube, Prof. S.S.Mane

dilipnawale27@gmail.com

Department of Computer,

BSP,Wagholi, Pune.

ABSTRACT

Our project is based on the Hadoop database file system. We will try to implement a framework for better and efficient replication. 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 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. Generally, maintaining the same replication mechanism for every data file leads to detrimental effects on the performance. By considering the drawbacks of the available system, we select the flexible system for replication. With the help of probability theory, the utilization of each data file can be predicted to create ac or responding replication strategy. Eventually, the popular files can be subs equently replicated according to their own access potentials. For the remaining low potential files, an erasure code is applied to maintain the reliability. Hence, our project will be improves the availability of the replicated file by considering the consistency and reliability. Furthermore, the complexity reduction is applied to enhance the effectiveness of the prediction when dealing with BigData.

ARTICLE INFO

Article History

Received: 14th March 2020

Received in revised form :

14th March 2020

Accepted: 16th March 2020

Published online :

17th March 2020

I. INTRODUCTION

The big data created a phenomenon in application and solution development to extract process and store useful information. It deals with new challenges. In big data analytic, Apache Hadoop is one of them ostren owned parallel frame works. Not only is it used to achieve high availability, Apache Hadoop is also designed to detect and handle consistency. Along with the development of Apache Hadoop, the Hadoop Distributed File System(HDFS) has been introduced to provide the reliability and high-throughput access for data-centric applications. HDFS is one of the suitable storage frameworks for parallel and distributed computing, especially for Map Reduce engine.

To improve the reliability ,HDFS is initially work by using mechanism that uniformly replicates three copies of every data file. This mechanism is majorly used for achieving fault tolerance. Keeping at least three copies makes the data more reliable when tolerating the failures.

The purpose of inventing Apache Hadoop was to achieve better performance in data processing and manipulation. The availability of cloud storage services is becoming a popular option for storing data that is accessible via a range of devices, such as personal computers, tablets, and mobile phones. Private cloud storage is a storage mechanism thatstores an organizations data at in-house storage servers by implementing cloud computing and storage

technology. Replication is a process wherea whole object is replicated some number of times, thus providing protection if a copy of an object is lost or unavailable and achieves fault tolerance feature. We implement a flexible replication management system to provide high availability for the data in HDFS, available data in HDFS improves the performance of the Hadoop system. We implement a complexity reduction method for the prediction technique by calculating the access potential for each file.

- **Scope of Project-**

This framework will be able to handle the big data. In the network, number of users are access data at a same time, so many times it create congestions. To overcome these problems, appropriate replication management is essential. We also try to use the separate data base for handling more big size of data.

II. METHODOLOGY AND IMPLEMENTATION

We implement the flexible replication system in HDFS to overcome the drawbacks of existing system. Here we deal with the replication system of the HDFS. We try to change its default strategy of replication.

Our framework is totally divided into three parts:-

1. Calculate access potential
2. Set replication fact or
3. Update replication of each file

Above three sections are helped to proposed Flexible Replication Framework.

Calculate Access Potential: Access potential is the term which used for calculating hit for the particular file. It gives the idea about the traffic of each file. Access potential is calculating the replication factor for each file. Access rate is depending on the number of hits in given unit of time. Data files are replicated using their own access potential. If we consider that, a specific file get total 72 hitsin4minute,the access potential for that file is 72 for 4 minute of time period. It means that, files get total 18 hits in a minute.

Replication Factor Calculation: The replication factor for each data files is calculated using the access potential of that file. For calculating replication factor, different parameters are used. Predication algorithm takes two parameters i.e.name of file and its own access potential value. After deciding the replication factor, the metadata is updated and replication is done in HDFS. Each node has some limited capacity for handling the traffic for each file. The capacity of node is depend on the RAM, processor and memory and storage space of that system.

Now, we take some parameters for calculating replication factor.

C=Capacity of the system to handle the traffic or hit for particular file. T = time in minutes
H = Number of hits in time T

Now, first we calculate the number of hits per minute,

No−hits−per−min = (no. of hits in time / T time in minute)
= H / T

Calculating the number of hits per minute, the replication factor for fies is finding out by referring the capacity C of system.

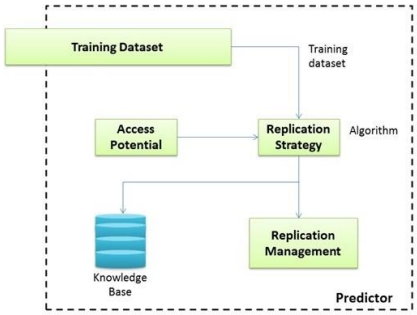
Therefore, Number of replications is,
No−replication= (No−hits−per−min / C)

According to default policy of Hadoop, replication factor is 3. So we add new replication factor in the existing and new replication factor is calculated.

New replication factor = 3 + No−replication

New replication factor denote the final replication factor of that particular file. Using this method, the replication factor for all existing file is calculated and store it in the meta data.

Updating Replication of Files: Now, after calculation of replication factor of each file, the replication of existing file is updated and new replication factor is set. Each file of HDFS is replicated with its new replication factor. After successful replication, each file is stored on different nodes of racks by following locality policy.



Goals and Objectives:

- To manage flexible replication factor of file
- To increase performance and maintain consistency
- To decrease unnecessary required time to access file

Purpose and Scope of Document:

This framework provides the better management of the Hadoop Database File System. It also maintains the consistency and reliability. Total framework is depending upon the replication factor of the files. So the scope of the project or framework is very vast.

Overview of responsibilities of Developer:-

- Therearedifferentresponsibilitiesofthedevelope rrelatedwiththeframework.
- In the flexible replication system, flexibility of the replication according to conditions is occurring. Framework helps to manage the replications on the appropriate nodes. So, developer has the responsibility of taking care about updating of database. Also, developer has responsibility of maintain connectivity with different applications.

Project Scope :

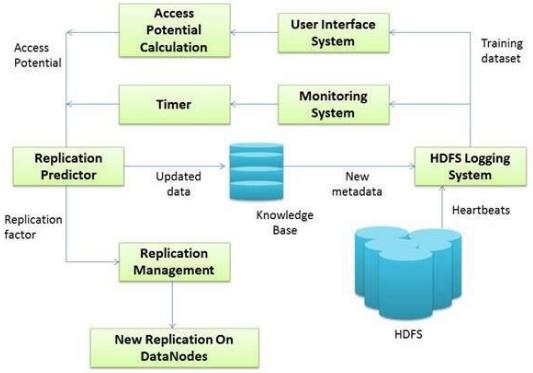
This framework will be able to handle the big data. In the network, number of users are access data at a same time, so many times it create congestions. To overcome these problems, appropriate replication management is essential. We also try to use the separate database for handling more big size of data.

III. SYSTEM DESIGN

In the architecture diagram, HDFS is the Hadoop Distributed File System which store number offiles in the format of block. Those files are access using the HDFS logging system. The data files are act as a training data in system. Size of these data may be in terms of megabyte (MB), gigabyte (GB), terabyte (TB) or petabyte(PB).

Training data is send to monitoring system and user interface system. Monitoring system is used to keep watch or take the feedback of system. Timer is used for set count-down for particular period in terms of minute. In that time period, access potential of each fileiscalculated.Whentimeisover,thecalculatedaccesspo tentialissendtothereplication predictor.

Replication predictor is used to set the replication factor for each file. When calculations are done, the information is send to knowledge base to update the metadata. On the other side, predicator sends updated information to replication management system. This system replicates the files on the nodes according to its new replication factor. The whole system is update with its new replication.



Algorithm -

1. Start server systems
2. Upload file in HDFS and create entry in database
3. Set count down
4. Calculate Access Potential
5. Calculate replication factor
6. Update database

7. Set new replication for each file

8. End

IV. CONCLUSION

In order 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 flexible to the characteristic of the data access pattern. The approach pro-actively per forms the replication in the predictive manner. Second, we implement 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 a flexible solution to advance the Hadoop system.

V. FUTURE SCOPE

The project is proposed from the need of the availability of the data on the nearest node from the requested user's node. Increase the locality of the frequently used files. Replicate the files depending on the access potential calculated for the requested file. Creating more replicas of the files with considering the availability and flexibility of the data.

References:

1. Adaptive Replication Management in HDFS based on Supervised Learning, by Dinh-Mao Bui, ShujaatHussain, Eui-Nam Huh, Sungyoung Lee,2015.

2. Enabling Efficient and Reliable Transition from Replication to Erasure Coding for Clustered File Systems, by Runhui Li, YuchongHu, Patrick P. C. Lee,2015.

3. Replication Management Framework for HDFS based on Prediction Technique, by Dinh-Mao Bui, Thien Huynh-The ,2015.

4. Multicast-based Replication for Hadoop HDFS, by JiadongWu and Bo Hong, 2015.

5. Fault Tolerant Erasure Coded Replication for HDFS Based Cloud Storage, by Aye Chan Ko,WintThidaZaw,2014.

6.Whatisapachehadoop?https://hadoop.apache.org/ ,accessed:2015-08-13.

7. R. Gallager, Stochastic Processes: Theory for Applications. Cambridge Univer- sity Press,2013.

8. C. Rasmussen and C. Williams, Gaussian Processes for Machine Learning, ser. Adaptive Computation And Machine Learning MITPress,2005. [Online].Available:http://www.gaussianprocess.org/gpml/chapters/

9. S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press,2004.

