Maintenance of File System And Improving Efficiency Of Hadoop By Using Clustering

Samandeep Kaur, Kewal Krishan

#Department Of Computer Science, Lovely Professional University
Phagwara, Punjab, India

Abstract-In the existing work, Hadoop distributed file system is defined as the large cluster set in which number of servers is collected for direct storage of data. The file system is defined as architecture for large dataset. In this present work we are basically maintaining the file system in form of clusters along with respective mount table definition. The mount table is attached with each cluster and the optimization of user query is been performed based on same mount table. The work is extended in two main phases, in very first phase the Hadoop architecture is defined in clusters. The cluster formation is an intelligent formation on keyword based feature analysis on files. The related files are kept in one cluster. Along with this, the mount table is defined that stores the keyword information as well as other metadata related to each file over the system. Once the architecture is defined, in the second phase the user query is filtered and the keyword is extracted from it. Based on the keyword analysis, the related cluster is selected and the query is performed on the selected cluster. The proposed work will improve the efficiency of the system.

——————————   ◆   ——————————

1 INTRODUCTION

Big Data is a term used to describe large collections of data (also known as datasets) that may be unstructured, and grow so large and quickly that it is difficult to manage with regular database or statistics tools, generally exceeding the processing capacity of conventional database systems. So, to gain value from this data, we must choose an alternative way to process it. With this massive quantity of data, businesses need fast, reliable, deeper data insight. Therefore, Big Data solutions based on Hadoop and other analytics software are becoming more and more relevant. The four V's are commonly used to characterize different aspects of big data which are as follows:-

1. Volume – Defined as the total number of bytes associated with the data.

2. Velocity – Defined as the pace at which the data are to be consumed.

3. Variety – Defined as the complexity of the data in this class.

4. Variability – Defined as the differing ways in which the data may be interpreted.[3]

1.1 Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware, similar with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highlyfault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides highthroughput access to application data and is suitable for applications that have large data sets.HDFS relaxes a few POSIX requirements to enable streaming access to file system data.[3]

1.2 HDFS Architecture

HDFS consists of interconnected clusters of nodes where files and directories reside. An HDFS cluster consists of a single node, known as a NameNode, which manages the file system namespace and regulates client access to files. In addition, data nodes (DataNodes) store data as blocks within files.[1]

1.3 Client Server Architecture

Client-server is a network architecture which separates the client (often a graphical user interface) from the server. Each instance of the client software can send requests to a server or application server.[1][2]

————————————————

- *Student, Samandeep Kaur is currently pursuing master's degree program in Computer Science and Engineering in Lovely Professional University, Phagwara, India. E-mail: deep.sam07@yahoo.com*
- *Assistant Professor, Kewal Krishan is currently working in department in Computer Science and Engineering in University, Phagwara, India,. E-mail: kewal.krishan@lpu.co.in*

1.3.1 HDFS clients

User applications access the file system using the HDFS client, a code library that exports the HDFS file system interface. HDFS supports operations to read, write and delete files, and operations to create and delete directories[2]. The user references files and directories by paths in the namespace. The user application generally does not need to know that file system metadata and storage are on different servers, or that blocks have multiple replicas.[1]

1.3.2 Hadoop System

Such networks are useful for many purposes. Sharing content files (see file sharing) containing audio, video, data or anything in digital format is very common, and real time data, such as telephony traffic, is also passed using Hadoop technology.

A pure Storage Area network does not have the notion of clients or servers, but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network.[1]

2. RELATED WORK

Philip H. Carns proposed a work on parallel file system under the linux-cluster based system to improve the performance of the system on Myrinet network in his paper.[6] In year 2004, **Jefferey Dean** proposed a work on the simplification of large cluster set (Google) by a programming model to process on large dataset effectively. The work is been implement with high degree of scalability.[4]**Sanjay Ghemwat** defined a File system for Google to design a large and scalable file system with high degree of aggregation.[7]

Work presented by **WittawatTantisiriroj** on internet services where the data oriented work is performed under the cloud environment as well as standard internet services. The work was designed for the standard file systems such as Google, Hadoop, and Amazon etc. The author performed the classification of the file system in the form of large clusters and presented a comparative analysis based on this parallel file system under the internet services. The author has used as service stack to present the Hadoop file system to analyse the performance of the system in an effective way. The author also provided the warehousing solution as the frame work to provide a support to the Hadoop application system.[8]

In Year 2009, **AshishThusoo** presented Hive, an open-source data warehousing solution built on top of Hadoop. Hive supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into map-reduce jobs executed on Hadoop. The facebook dataset called hive warehouse is been used to handle the tera byes of the datasets.[9]

In Year 2009, **Sage A. Weil** performed a work," Ceph: A Scalable, High-Performance Distributed File System". Author has developed Ceph, a distributed file system that provides excellent performance, reliability, and scalability.[5]

3. Present work

3.1 Scope of Study

The descriptive data is always presented in the form of documents and these documents exist in different file formats. When the work is performed for particular enterprises, it contains a vast collection of files over the system. In such case the management of these files and handling the file system query is itself a challenging task. In this present work, Hadoop-based architecture is presented to define file system architecture. The presented work is quite beneficial as:

1. As the work is based on cluster based, it reduces the size of database query. Instead of maintaining the complete file system individually, the management of the specific clusters is easy to represent.

2. The cluster definition enables the easy migration of the sub-file system on different location physically.

3. While working on distributed system, such kind of architecture is more beneficial as it can maintain the cluster location wise.

4. As the cluster formation is keyword based, the query analysis easily identify the required cluster.

5. As the mount table is maintained for each cluster to maintain the cluster data, the query processing will be more effective.

3.2 Problem Formulation and proposed solution

A file system is one of the most traditional and widely used mechanisms to maintain the user data in the form of distributed system. The problem in existing file system is that as the data increases it is hard to maintain it and whenever the client does any query it has to process all the data and it takes so much time to fetch the result of the query. To overcome this problem wehave proposed an model or approach which overcome this problem efficiently. In this paper, we have proposed Hadoop based file system architecture basically designed for large file dataset where we have terabytes or petabytes of data in the form of files and need to avail the information to user effectively on request. The work is defined in two stages. In first stage, the file system architecture is defined and in second stage, the effective user query is defined. To maintain the data effectively a clustered file system is defined. Complete file system will be divided in the form of clusters and the cluster definitions are based on keyword analysis over the system. Each cluster separately maintains a mount table to keep track of the files presented in the system. As the user will pass a query to the Hadoop system, at first the keyword extract from the query will be performed. Based on the keyword based match, the relative cluster will be identified. Now to retrieve the relative file content, a search will be implemented on mount table that contains the descriptive information along with location specification for each file of the cluster. From this mount table search the actual path of the related contents will be shown and it will also generate the download information in terms of time taken, load etc. The presented application will also define the modules in terms of file up loader and its maintenance under the concept of Hadoop. The database management will be performed based on keyword analysis in the form of mount table. The query processing is performed in two steps, first to identify the cluster and second to identify the file location and other information within cluster.

3.3 Proposed Framework

To solve the problem of distributed file system processing a novel clustered approach is suggested, as shown in Figure. The user sends a request to global query interface of distributed file system; the query is received by query analyzer which analyzes the query with the help of global schema. Query analyzer breaks the query into sub-queries and sends to cost optimizer for cost estimation of sub queries. Cost optimizer analyzes the cost with the help of data dictionary. Query distributor has the main responsibility to receive sub-query from cost optimizer and sends to appropriate local optimizer of local site of cluster. A mount table is placed between local cache and database. Mount tables are like the index of the book which contains all the information of the files like location and extension. Whenever any client does the query first the local optimizer or system see in the mount table its index and directly go to that data and fetch it. It will increase the speed of the processing file system.

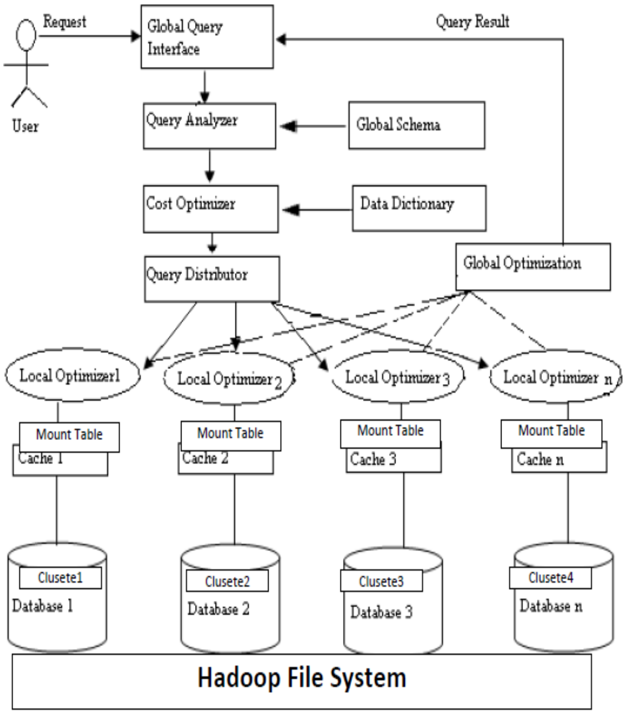


Figure: Proposed Architecture

4. Conclusion

In this paper we go through the HDFS architecture and different model of the HDFS that are purposed by different researcher to handle the big data. We have proposed a approach to increase the processing speed of the HDFS file system by placing the mount table between local cache and data base that will increase the query processing of the system. Mount contains the track of files or information in it of the cluster so that query processing will be fast. By doing this method we avoid the extra processing of the database in cluster.Our idea is to make the existing system from efficient and faster.

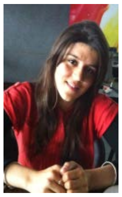
ACKNOWLEDGMENT

I would like to acknowledge our respected mentor Ms. Sheena Singh (Assistant Professor, LPU) for their valuable time, suggestions which are responsible for the work produced here. Many thanks to the anonymous referees also for helping immensely to improve this paper.

References

[1] KonstantinShvachko,Kuang, Sanjay Robert chansler "The Hadoop Distributed File System" IEEE 2010.
[2] Hadoop.http://www.apache.com
[3]BigData.http://www.bigdatauniversity.com
[4]Jeffrey Dean (2004)," MapReduce: Simplified Data Processing on Large Clusters", USENIX Association OSDI '04: 6th Symposium on Operating Systems Design and Implementation pp 137-149

[5]Sage A. Weil," Ceph: A Scalable, High-Performance Distributed File System".
[6] Philip H. Carns (2000)," PVFS: A Parallel File System for Linux Clusters", In Proc. of the Extreme Linux Track: 4th Annual Linux Showcase and Conference, October 2000.
[7]Sanjay Ghemawat (2003)," The Google File System", SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA
Mengwei Ding (2011)," More Convenient More Overhead: The Performance Evaluation of Hadoop Streaming", RACS '11, November 2-5, 2011, Miami, FL, USA. pp 307-313
[8]WittawatTantisiriroj," Data-intensive _le systems for Internet services: A rose by any other name ...".
[9] AshishThusoo(2009)," Hive A Warehousing Solution Over a MapReduceFramework", VLDB '09, August 2428, 2009, Lyon, Fran



Samandeep Kaur received her B.Tech degree in Computer Science and Engineering from CTIEMT, Jalandhar (PTU), and Punjab, India in 2011, now she is doing M.Tech in Computer Science and Technology from Lovely Professional University, Phagwara, and Punjab, India. Her research interests Databases and Data Mining.

K**ewal Krishan** received his B.Tech degree in Computer Science and Engineering from Amaravati University,Maharashtra 1994, And M.Tech in Computer Science and Engineering from Punjab Technical Univ. in 2007 . Now he is working as an Assistant Professor with Department of CSE at Lovely Professional University, Phagwara, and Punjab, India.