INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 1)
Available online at www.ijariit.com

Big Data Analysis Using Apache Hadoop

Suruchi Padhy
Cambridge Institute of Technology,
Bangalore, Karnataka
padhysuruchi@yahoo.in

Dr. Shashi Kumar D R
Cambridge Institute of Technology,
Bangalore, Karnataka
hod.cse@citech.edu.in

ABSTRACT

Traditional data management, warehousing, and analysis systems fall short of tools to analyze this data. Using traditional DBMS techniques like Joins and Indexing and other techniques like graph search is tedious and time consuming.
In this paper, we suggest various methods for catering to the problems in hand through Map Reduce framework over Hadoop Distributed File System (HDFS). Map Reduce is a Minimization technique which makes use of file indexing with mapping, sorting, shuffling and finally reducing. Map Reduce techniques have been studied in this paper which is implemented for Big Data analysis using HDFS.

Keywords: *Big Data, Apache Hadoop, HDFS, MapReduce.*

1. INTRODUCTION

Big data consists of a heterogeneous mixture of data structured and unstructured data." Big data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze. Manipulation of large datasets possesses problems of computational speed and error recovery.

In a distributed data system, resources and cost of maintenance can scale up as per need and the proposed system is flexible enough to handle this. Also, the framework used is intelligent enough to handle errors on its own. When coupled with the advantages of parallel processing, we are able to get a faster system.

The parallel processing problem can overcome by using Apache Hadoop which is designed to scale up from a single server to lots of machines each offering local computation as well as.

2. RELATED WORK

Data management and reporting are very important for any organization to make decisions. Retrieving data from traditional database system not only time consuming and retrieving graphical and storing unstructured data and retrieving has limitation. For analysis of big data, database integration and cleaning is much harder than the traditional mining approaches. This can be addressed by distributing the computation over several nodes each of which works in parallel on a subset of the complete dataset and maintains coherence for producing an appropriate result. Also, the whole system is able to handle data coming from different sources which can be in different formats. Data size grows with the number of computational units present on the system.

Hadoop can help to overcome these limitations. There are many related works in the literature about Hadoop and data computing and key technologies to achieve the vision of Hadoop computing. The author introduces an analysis of unstructured data, con and MapReduce services, and explains how these components can interact to produce a better analysis experience. The main goal of this work is to show the feasibility of such implementation, introducing a new partition scheme for tasks. The best point about this paper is the considerations about using the Hadoop MapReduce computing. They propose the creation of high end computing clusters to run applications/services the same way that they will run on different desktop devices in order to avoid inconsistencies produced to run part of a program in HDFS architecture.

3. HDFS ARCHITECTURE

HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file.

Files in HDFS are write-once and have strictly one writer at any time.

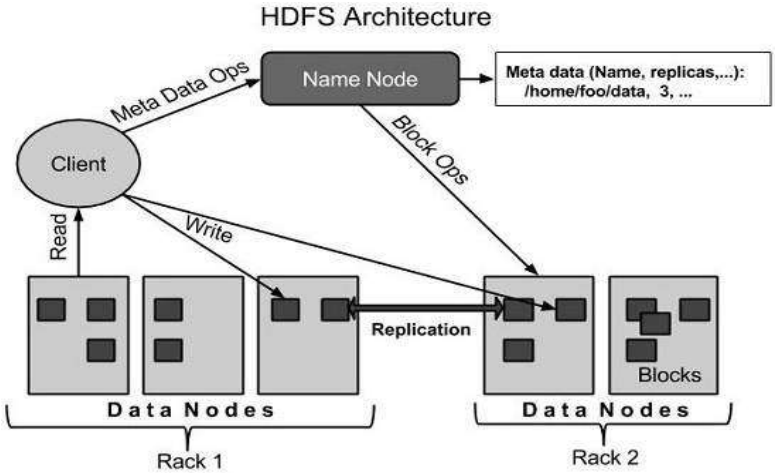

Fig 1 HDFS Architecture

HDFS follows the master-slave architecture and it has the following elements.

Name Node: The name node is the commodity hardware that contains the GNU/Linux operating system and the name node software. It is software that can be run on commodity hardware. The NameNode makes all decisions regarding replication of blocks. It periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. The system having the name node acts as the master server and it does the following tasks:

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Data node: The data node is a commodity hardware having the GNU/Linux operating system and data node software. For every node (Commodity hardware/System) in a cluster, there will be a data node. These nodes manage the data storage of their system. Data nodes perform read-write operations on the file systems, as per client request. They also perform operations such as block creation, deletion, and replication according to the instructions of the name node.

Block: Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Figure 7.2 shows the architecture of HDFS clusters implementation with Hadoop. It can be seen that HDFS has distributed the task over two parallel clusters with one server and two slave nodes each. Data analysis tasks are distributed in these clusters.

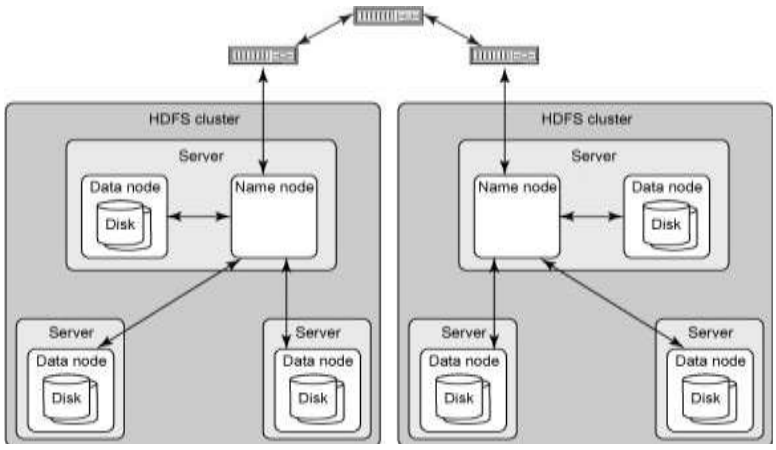

Fig 2 HDFS Clusters

At its core, Hadoop has two major layers namely:

- Processing/Computation layer (MapReduce), and
- Storage layer (Hadoop Distributed File System).

MapReduce

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.

Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules:

- **Hadoop Common**: These are Java libraries and utilities required by other Hadoop modules.
- **Hadoop YARN**: This is a framework for job scheduling and cluster resource management.

In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.

4. CONCLUSION

Big Data analysis tools like MapReduce over Hadoop and HDFS, promises to help organizations better understand their customers and the marketplace, hopefully leading to better business decisions and competitive advantages.

MapReduce can be exploited to solve a variety of problems related to text processing at scales that would have been unthinkable a few years ago.

5. REFERENCES

[1] HDFS: Available: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[2] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Communications of the ACM, 51 (1): 107-113, 2008
[3] Apache hadoop, http://hadoop.apache.org
[4] Tom White, O'REILLY, "Hadoop – The Definitive Guide"
[5] Java MapReduce http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
[6] https://en.wikipedia.org/wiki/MapReduce

