ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

Predictive Analysis Using Hadoop: A Survey

Shreyas Kudale[1], Advait Kulkarni[2], Asst. Prof. Leena A. Deshpande[3]

Student, Department of Computer Engineering, VIIT, Pune, India

Student, Department of Computer Engineering, VIIT, Pune, India

Assistant Professor, Department of Computer Engineering, VIIT, Pune, India

ABSTRACT**:** Current buzzword in the IT industry is of Big Data. But what exactly is "Big data"? Any amount of data which becomes difficult to process by using traditional RDBMS can be referred to as Big Data. Data is being considered to be the future asset of today's organizations. Organizations from the public and private sector are making a strategic decision to use this data generated to gain competitive advantage. The main hurdle is to process this huge data efficiently for analytics purpose.

Analysis of such huge data to obtain information out of it by the traditional relational database model (RDBMS) is costly as well as inefficient. The use of Hadoop framework can be made for cost effective and faster data processing, which would enhance the prediction process. Through this paper, we suggest the use of Hadoop Framework and the E-T-L process for Hadoop for performing predictions based on the datasets. Basic introduction on use of Apriori algorithm on Hadoop for data analysis is also presented**.**

Keywords: Big Data, RDBMS, HDFS- Hadoop Distributed File System, Map Reduce, E-T-L-Extract Transform and Load**.**

I. Introduction

With the explosion in user-generated content from social media and organizations, enormous amount of data is being generated every day. Petabytes of data is produced each day and to store such huge amount of data in a cost effective way is a daunting task. An example of this is that, [6] Twitter processes over 7 terabytes of data per day, Google handles 24 terabytes of data per day and the AT&T handles around 19 terabytes of data in a single day. Every 60 seconds, a tidal wave of unstructured data get produced, consumed and archived. All this data growth implies that big data and real time analytics is a big focus in social and mobile data along with business organizations. This data is an asset to any working organisation and is being thought of as the fundamental thing, which will change how their business works. Hence the focus is more on the analysis of this enormous amount of data for future predictions, growth strategies for the organizations. This gave rise to the analytics and prediction domain in data mining sphere. In today's times we require fast and readily available information which can only be achieved by very high processing speed and data storage capacity. As most of the data coming from web is unstructured, storing and processing of this data should be cost effective to be used by organizations. Traditional Information technology architecture falls short and over burdened to handle such large data. Apache Hadoop is emerging as the obvious choice for managing big unstructured data. For example EBay –online shopping site used Hadoop to improve search quality, user experience and feedback analysis. The Hadoop framework came into picture in 2004, which provided an easy and reliable implementation of distributed computing. Also the MapReduce algorithm used by Hadoop made calculations and processes easier to run parallel on many computers. The Hadoop framework unleashed the real power of distributed computing. It thus provides analytic decision-making data for businesses of any size.

II. Motivation

A. Shortcomings of traditional RDBMS

As majority of data is unstructured, use of relational model for data storage is an inappropriate choice as converting the unstructured or semi-structured data into relational and structured form is a non economical and costly job. Data should be homogenous and should have a specified schema to be used in a relational database .RDBMS is an overkill and expensive option for data storage when compared to Hadoop. Fault tolerance is low in case of RDBMS. Efficiency

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

of traditional RDBMS is less in contrast to Hadoop. [10] Though RDBMS is still used for real-time and relational data, but only when data is not large compared to raw/content data.

Due to expensive data storage costs in RDBMS, organizations store only part of data for prediction purpose. While using Hadoop one can keep all the data needed in future .Considering the heterogeneous structure and multiple architectures of the big data, RDBMS falls short to meet all its need.

III.HADOOP ARCHITECTURE

A. MAPREDUCE PARADIGM

Apache Hadoop is an open source software platform for storing and processing data. It is written in Java, it runs on clusters of industry standard servers configured with direct attached storage. Petabytes of data can be reliably stored on tens of thousands of servers while scaling performance cost effectively by merely adding inexpensive nodes to the cluster. Apache Hadoop uses the processing framework known as MapReduce. MapReduce helps programming solve data parallel problems for which data set can be sub-divided into small parts and processed independently. Ref [1]The system splits the input data into multiple chunks, each of which is assigned a map task that can process the data in parallel. Each map task reads the input as a set of (key, value) pairs and produces a transformed set of (key, value) pairs as the output. The framework shuffles and sorts outputs of the map tasks, sending the intermediate (key, value) pairs to reduce task, which groups them into final results. MapReduce uses JobTracker and TaskTracker mechanisms to schedule tasks, monitor them and restart any that fail.

Hadoop framework utilises its extensive API based on map reduce programming model to extract and analyse data from semi-structured and unstructured data with elegance and very high efficiency by providing for the users to use their own mapping and reducing algorithms. MapReduce is an important advance because it allows ordinary developers, not just those skilled in high-performance computing, to use parallel programming constructs without worrying about the

Complex details of intra-cluster communication, task monitoring, and failure handling.

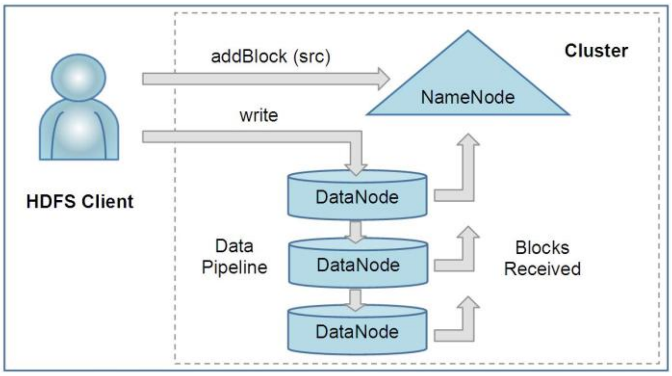

Figure 1: HDFS client write operation [1]

B. HADOOP DISTRIBUTED FILE SYSTEM

1) *NameNode:* The Apache Hadoop platform also includes the Hadoop Distributed File System (HDFS) which is designed for scalability and fault tolerance. [1] HDFS stores large files by dividing them into blocks (usually 64 or 128 megabytes) and replicating the blocks on three or more servers. Maintaining the namespace tree and the mapping of file blocks to DataNodes (the physical location of file data) is done by the NameNode. Any HDFS client who wants to read a file must first contact the NameNode to acquire the locations of data blocks comprising the file and then from the DataNode closest to the client, should read the block contents. While in case of writing, data, the client makes request to the NameNode to allot a suite of three DataNodes to host the block replicas. HDFS stores the entire namespace in RAM.

The list of blocks and inode data belonging to each file contains the metadata of the name system called as the image. The permanent record of this image is stored onto the local host's native file system called as the checkpoint. Storage of the modified log of the image called as the journal in the local host's native file system is also performed by the NameNode. For durability purpose, the checkpoint's redundant copies and that of journal can be formed at other servers. HDFS provides API's for MapReduce applications to read and write data in parallel. Capacity and performance can be scaled by adding DataNodes and a single NameNode mechanism manages data placement and monitors server availability. HDFS cluster in production use today reliably hold Petabytes of data on thousands of nodes.

2) *DataNodes:* Each data block replica on a DataNode comprises of two files in the native file system of local host. The data is in the first file and the block's metadata consisting checksums for the block data and the block's generation stamp is in the second file.

At start up each DataNode connects to the NameNode to perform a handshake to check the namespace ID and the software version of the DataNode. The namespace ID is assigned to the file system instance when it is formatted. All nodes of the cluster store the Namespace ID persistently. Nodes having a different namespace ID will not be able to join the cluster, thus maintaining the integrity of the system. During course of normal operation DataNode sends heartbeats to the NameNode so as to confirm that DataNode is operating and availability of the block replicas it hosts.

3) *HDFS Client:* HDFS file system can be accessed using the HDFS client, which is a code library exporting the interface of HDFS file system. Operations like reading, writing, deleting files, and operations for creation and deletion of directories are supported by HDFS. User applications refer files and directories by paths in the namespace.

An API provided by HDFS exposes the locations of a file block that allows applications like the MapReduce framework in scheduling a task to where the data are located, inturn improving the read performance.

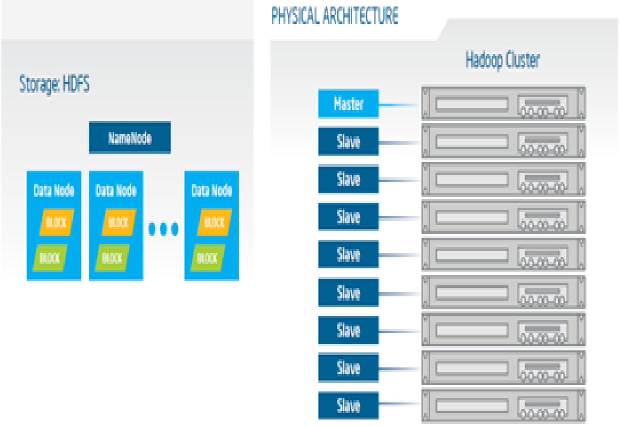

Figure 2: HDFS Architecture [6]

4) *Image and journal:* Metadata of the file system that describes the organization of application data in form of directories and files is called as Namespace image. Checkpoint is a permanent record of the image written to disk. A write-ahead commit log for changes made to the file system that are permanent is known as Journal. Each time a client-initiated transaction occurs, it is recorded in the journal.

5) *CheckpointNode:* NameNode in HDFS also performs the role of a CheckpointNode or a BackupNode which is specified at the startup of node.

The function of combining the existing checkpoint and journal for creation of a new checkpoint and an empty journal is performed by the CheckpointNode. It retrieves the journal files and current checkpoint from the NameNode, merges them logically and returns the new checkpoint to the NameNode.

6) *BackupNode:* New feature of HDFS is the BackupNode which is capable of creating periodic checkpoints and also maintains an up-to-date, in-memory data image of file system namespace which is always synchronized with the state of the NameNode.

The BackupNode can be viewed as a read-only NameNode which contains all file system metadata information except the block locations. All regular functions of NameNode except modification of namespace are performed by BackupNode. Option of running the NameNode without persistent storage is also provided by this node.
Following diagram depicts the overall working of Hadoop sysytem .

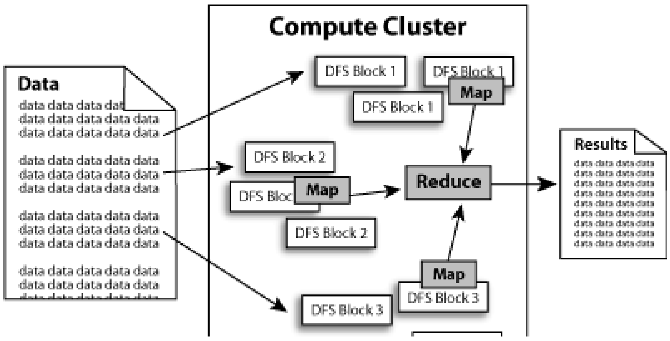

Figure 3: Overview of Hadoop [3]

IV. PREDICTIVE ANALYSIS USING HADOOP

A. ETL USING HADOOP

The process of extracting data from multiple sources, transforming it to fit individual analytical needs, and loading into a data storage is called as “Extract, Transform and Load” (ETL). The nature of big data is such that it requires the infrastructure for this process to scale cost-effectively. Hence Apache Hadoop is emerging as the suitable option for managing big data.

ETL tools move data from one place to another by performing three functions: [6]

1) Extract data from sources such as ERP or CRM applications.

During the extract step, one may need to collect data from several source systems and in multiple file formats, such as flat files with delimiters (CSV) and XML files. One also needs to collect data from legacy systems that store data in arcane formats no one else uses anymore.

2) That data is transformed into a common format that fits other data in the warehouse. Steps like multiple data manipulations, such as moving, splitting, translating, merging, sorting, pivoting, and more come under transform process,

3) Load the transformed data into the data warehouse for analysis. This step can be performed in batch processes or row by row.

Much has changed in data warehousing over the past two decades. Chiefly, databases have become vastly more powerful. None of the traditional ETL solution is cheap and their cost and complexity increases with big data.

Hadoop brings at least two major advantages over traditional ETL:

1) Ingest massive amounts of data without specifying a schema on write. A key characteristic of Hadoop is called “no schema-on-write,” which means you do not need to specify the data schema prior to loading of data onto Hadoop. This is true not only for structured data (such as point-of-sale transactions, call detail records, and call centre transactions), but also for unstructured data (such as comments of users, doctor’s notes, insurance claims descriptions, and web logs) and social media data (from sites such as Facebook, LinkedIn, and Twitter). Regardless of whether the incoming data has explicit or implicit structure, you can rapidly load it as-is into Hadoop, where it is available for downstream analytic processes.

2) Offload the transformation of raw data by parallel processing at scale. Once the data is in Hadoop (on a Hadoop-compatible file system), one can perform the traditional ETL tasks of cleansing, normalizing, aligning, and aggregating data by employing the massive scalability of MapReduce. Hadoop allows you to avoid the transformation bottleneck in the traditional ETL by off-loading the ingestion, transformation, and integration of unstructured data into the data storage like warehouse. Because of its scalable performance, Hadoop significantly accelerate the ETL jobs. Data stored in Hadoop can persist over a much longer duration, hence helping in analysis purpose.

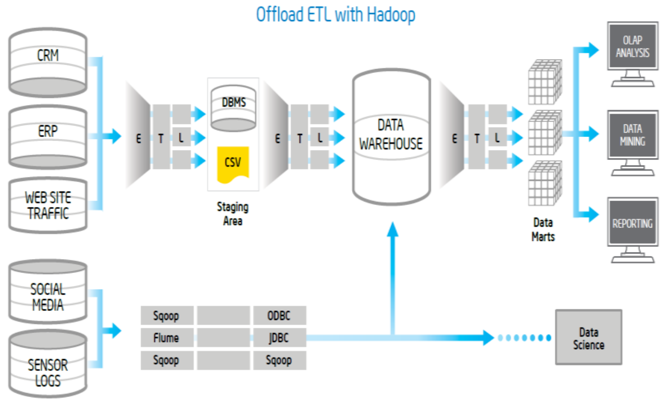

Figure 4: ETL with Hadoop [6]

B. HADOOP OVER TRADITIONAL DATA MINING

Discovery from large databases and extraction of hidden information is called as data mining. Its application areas are decision support, forecasting, estimation, etc.

Web Data mining is the scanning of large amounts of data to find a hidden regularity found in the contents of the application to solve data quality issues. Many of the available data mining algorithms can work on data having a fixed structure, where in advance the data scheme is defined .But web data often does not have regular structure. The data generated from the Internet, or from the private Intranets of many companies, has multiple structures. Structure of data is priory known in traditional relational or object-oriented databases. On the other hand data from the web is mainly unstructured, consisting of images, sounds, and raw text. As majority of the data falls somewhere in between these two paradigms, for many reasons: the data may be structured, but the structure is not known to the user or many times user ignores the structure for purpose of browsing. Structures may be sometimes implicit, like that of formatted text or may be opposite to traditional databases which are rigid and regular. So use of Hadoop provides a solution in processing of such unstructured data.

C. ASSOCIATION RULE MINING USING HADOOP

Apriori, one of the most important algorithms for data mining is a subpart of Association rule mining technique. Association rule mining [2] is to find rules in the database with minimum support and minimum confidence which are given by the user. Many improved versions of Apriori algorithm like Apriori using Boolean matrix on Hadoop have evolved to increase efficiency of data mining process. With the use of such efficient techniques for data mining, prediction analysis is developing in terms of accuracy and time complexity.

V. CONCLUSION

We gave an introduction on how the Hadoop framework can be used for large data storage and analytics purpose through this paper. Large amount of source data from social media, web logs or third party stores is stored on Hadoop to enhance analytic models that drives research and discovery. Data can be stored on Hadoop clusters in cost effective manner and can be retrieved easily when needed. Operational cost of whole data analytics and data processing can be lowered by use of Apache Hadoop. Its MapReduce on HDFS provides a scalable, fault tolerant platform for processing large amount of heterogeneous data.

Organizations gain an additional ability to store and access data that they need without storing such data onto warehouses. Hadoop is not an ETL tool but a platform that supports running ETL processes in parallel. As corporations start using larger amounts of data, migrating it over the network for transformation or analysis becomes unrealistic. Moving all the big data to one storage area network (SAN) or ETL server becomes infeasible with big data volumes. . With Hadoop, raw data is loaded directly to low cost commodity servers one time, and only the higher value refined results are passed to other systems. Rapidly ingesting, storing, and processing big data requires a cost effective infrastructure that can scale with the amount of data and the scope of analysis. When the source data sets are large, fast, and unstructured, Apache Hadoop is the convenient and feasible option for data analysis.

REFERENCES

[1] K.V.Shvachko,"The Hadoop Distributed File System Requirements," MSST '10 Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)
[2] Lan Huang; Wang Xiao-wei; Zhai Yan-dong; Bin Yang, "Extraction of User Profile Based on the Hadoop Framework," *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on* , vol., no., pp.1,6, 24-26 Sept. 2009
[3] Honglie Yu, Jun Wen, Hongmei Wang ,Li Jun ," An Improved Apriori Algorithm Based On the BooleanMatrix and Hadoop" , Procedia Engineering, Volume 15, 2011, Pages 1827–1831
[4] Dhruba Borthakur. "The Hadoop Distributed File System:Architecture and Design",http://lucene.apache.org/hadoop/hdfs.html
[5] J. Venner, Pro Hadoop. Apress, June 22, 2009
[6] Bringing the Power of SAS® to Hadoop .Combine SAS® World-Class Analytic Strength with Hadoop's Low-Cost, High-Performance Data Storage and Processing to Get Better Answers, Faster.
[7] White paper on big data analytics by Intel "Extract, Transform, Load Big Data using Apache Hadoop"
[8] SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets, Ronnie Chaiken, Bob Jenkins, Per-Åke Larson, Bill Ramsey, Darren Shakib, Simon Weaver, Jingren Zhou, Microsoft Corporation {rchaiken, bobjen, palarson, brams, darrens, sweaver, jrzhou}@microsoft.com
[9] Big Data: Hadoop, Business Analytics and Beyond: http://wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond
[10] Hadoop over RDBMS: http://www.computerworlduk.com/in-depth/applications/3329092/hadoop
[11] Big Data: http://bigdataprocessing.wordpress.com
[12] Apache Hadoop: http://hadoop.apache.org

BIOGRAPHY

Name: **Shreyas S Kudale**
Designation: Final year student of BE (Computer engineering) in Vishwakarma Institute of Information Technology, Pune-411048, Maharshtra, India. Area of interest: Data mining and Hadoop. BE Project : Efficient data mining techniques.

Name: Advait S Kulkarni
Designation: Final year student of BE (Computer engineering) in Vishwakarma Institute of Information Technology, Pune-411048, Maharshtra, India. Area of interest: Hadoop and distributed computing. Project : Efficient data mining techniques.