International Conference On Emanations in Modern Technology and Engineering (ICEMTE-2017) ISSN: 2321-8169
Volume: 5 Issue: 3 66 - 68

Handling Big Data using Hadoop

Nishigandha Karbhari
Department of Computer Engineering
Shree L.R. Tiwari college of engineering
Mira Road, Mumbai, India
nishi_karbhari@yahoo.com

Dr. Vinayak D. Shinde
Department of Computer Engineering
Shree L.R. Tiwari college of engineering
Mira Road, Mumbai, India
vdshinde@gmail.com

Abstract – We are at the era where everything around us is computerized therefore creates a large amount of data. This data is of all kind and needs processing and analysing to get an outcome from it. Big data analytics refers to the process of collecting, organizing and analyzing very large sets of data to discover patterns and other useful information. The present paper proposes the concept of map reduce data mining tool with Hadoop functionalities. Big data analytics need to be combined with other process to increase effectiveness and offer pioneer services to customers. Business include various big data analytics for this process. In this paper we discuss how Big Data analytics can deliver the benefits to organizations and the various aspects to work with big data.

Index Terms – *Big Data, Map reduce, hadoop.*

*****

I. INTRODUCTION

"Big data exceeds the range of frequently used hardware environments and software tools to capture, manage and process it within a tolerable elapsed time for its user population.".[1] Big data is data that exceeds the handling capability of any conventional database systems. The data is too big, moves too fast, or does not fit the structures of traditional database architectures. To increase value from this data, you must choose an alternate procedure it, i.e., effective data analytics is required. This module gives you a complete picture about big data and how analytics is carried on such a data. Data Analytics (DA) is the science of examining raw data with the purpose of drawing conclusions about that information. The data that is captured by any data collection agent or tool or software is in its raw form, i.e., unformatted or unstructured or unclean with noises/errors or redundant or inconsistent.

Big data can be described by three major characteristics Volume, Variety and Velocity. [3] Volume is the quantity of generated data, important in this context. The size of the data determines the value and potential of the data whether it can actually be considered big data or not. The name big data contains a term related to size, and hence the characteristic. Variety is the type of content, and an essential fact that data analysts must know. It helps people who associate with and analyse the data to effectively use the data to their advantage and thus uphold its importance. Velocity is speed at which the data is generated. It is processed to achieve demands and challenges that include in the path of growth and development.

Big data analytics can be used by data scientists to analyse huge volumes of data. Considering that your organization could accumulate multiple row or consider billions of rows. With data of hundreds of millions of data combinations in multiple data stores and ample formats. High-performance analytics is necessary to process that much data in order to figure out what's important and what isn't. Analysing big data allows analysts, researchers, and business users to make better and faster decisions using data that was previously inaccessible or unusable. Using analytics techniques example text analytics, machine learning, predictive analytics, data mining, statistics, and natural language processing, businesses could analyse previously untapped data sources independent or collected with their existing enterprise data to gain new visions resulting in meaningfully better and faster decisions.

II. BIG DATA

The recent hot IT buzzword, big data has become viable as economic methods have developed the volume, velocity and variability of enormous data. Data included valuable patterns and information, previously hidden for of the amount of work required to extract them. Big data exceeds processing capacity for conventional database systems. The data is too big, moves too fast, or doesn't fit the structures of traditional database architectures. To gain value from this data, you must choose an alternative way to process it. "Big data" can be pretty unclear, in the same way as the term "cloud" covers varied technologies. Input data to big data systems could go on from social networks, web server logs, traffic flow sensors, satellite imagery, broadcast audio streams, banking transactions, MP3 of music, the content of web pages, scans of government documents, GPS trails, telemetry from automobiles, financial market data, etc. Quantum of data processed: [1]

•The Internet Archives passed 15 petabytes as of May 2014
•Google processes 30 PB a day-"MapReduce"-Portal.acm.org
•AT & T transfers about 30 petabytes of data through its networks each day -"AT&T-News Room"
•The German Climate Computing Centre (DKRZ) has a storage capacity of 60 petabytes of climate data
•Wayback Machine has 3PB + 100 TB/month
•Facebook has 2.5 PB of user data + 15 TB/day
•As of January 2013,Facebook users had uploaded over 240 billion photos. For each uploaded photo, Facebook

generates and stores four images of different sizes, which translated to a total of 960 billion images and an estimated 357 petabytes of storage
•eBay has 6.5 PB of user data + 50 TB/day
•CERN's Large Hydron Collider (LHC) generates 30 PB a year.
•Movie Avataris reported to have taken over 1 petabyte of local storage for the rendering of the 3D CGI effects
•Teradata Database 12 has a capacity of 50 petabytes of compressed data.

Typically the following points may be used by the readers to understand how big data differs from that of a traditional data which is organized in single or multiple storages:
•Generated automatically by machine.
•Not designed to be friendly.
•Can be messy (junk filled data).
•No standards

III. HADOOP DATA ANALYTICS TECHNOLOGIES

3.1 HIVE / HIVEQL

Hive is a data warehouse package built on top of Hadoop to process structured data. It is a Hadoop runtime support structure that allows leverage the Hadoop platform with SQL which is a common relational database. In Hive the developers write the Hive Query Language (HQL) which is similar to the standard SQL statements. It includes less commands which can be useful in the map-reduce technique across Hadoop clusters.

Hive is not a relational database or an On-line transaction processing (OLTP). [5]Hive is online analytical processing which provides SQL type of language for querying. It is commonly known as HiveQL or HQL. Hive chooses respective database server to store metadata of tables or database, their datatypes and HDFS mapping.

3.2 PIG / PIGLATIN

Pig developed at Yahoo! was to allow people using Apache Hadoop® to focus more on analyzing large data sets. To reduce time to write mapper and reducer programs. The Pig programming language is designed to handle any kind of data. It is of two components first is the language, PigLatin and the second is a runtime environment. [4]
The programming language:
1. The Pig program LOADS the data which is to be manipulated from HDFS.
2. Then run the data through a set of transformations. Transformation is translation of the mapper and the reducer task.
3. Last, DUMP the data to the screen and STORE the results in a file somewhere.

3.3 HBASE

HBase is a column-oriented DBMS which runs on top of HDFS. It is used for sparse data sets, which are a common aspect in big data use cases. Its applications are written in Java similar to typical MapReduce application. [4]It contains a set of tables with rows and columns, like a traditional database. Each table have a Primary Key, and all access is done using it. The column represents an attribute of an object and allows for many attributes to be grouped together known as column families, which are all stored together.

HBase is called the Hadoop database because its NoSQL feature and database that runs on top of Hadoop. It includes scalability of Hadoop by running on the Hadoop Distributed File System, with real-time data access as a key/value store. It is meant to host huge tables with billions of records/rows with millions of columns to run across a cluster. It allows you to query for individual records as well as derive aggregate analytic reports across an enormous amount of data.

3.4 MAPREDUCE

Map reduce is implemented HDFC. Hadoop framework allows the processing of large data set across clusters of computer. It is distributed processing to scale up single server to thousands of server consisting of large computing and storage. [2] HDFS used by Hadoop application is a primary storage system. It provides high performance data access to the clusters in Hadoop. on MapReduce is a programming model that can process Big Data in parallel on multiple nodes which is used for writing applications. MapReduce includes analytical skills for evaluating huge capacity of complex data. It is a parallel programming model which is used to process large amount of data on various clusters of the Hadoop for processing unstructured, semi structured and structured data. HDFS has a Name Node and Slave Nodes, whereas MapReduce has Job Tracker and Task Tracker slaves.[2]

The centralized system creates bottleneck while processing various files at a single intense. Google solved the bottleneck issue using MapReduce algorithm. MapReduce divides a task into small parts and allocate it to multiple computers. Then, these results are collected and combined to form the final result dataset. It is an associated implementation for processing and generating large data sets.[6] Users have a map function to processes a key/value pair and which generate a set of intermediate key/value pairs, a reduce function is used to merge all intermediate values associated with the same key. Many real world tasks are expressible in this model. Programs written in this are automatically parallelized and executed on a large cluster of product machines.

There are two component on which the map reduce functions job tracker and task tracker. Job Tracker in Hadoop is service to guide the map reduce task to point precise data node with the desired data required to process through map reduce task. A Task Tracker is a node to accept tasks for Map, Reduce and Shuffle operations from a Job Tracker. It is configured with slots to indicate the number of tasks that it can accept. The Task Tracker observers these processes, capturing the output and exit codes. When the process completes, successfully or not, the tracker reports the Job Tracker.
The MapReduce algorithm contains two important tasks:
- The map task is done by means of Mapper Class.
- The reduce task is done by means of Reducer Class.

3.4.1 Mapper Class

The Mapper class is used to define the Map job. Maps input key-value pairs to a set of intermediate key-value pairs. Maps are the individual tasks that converts the input records into

intermediate records. The converted intermediate records need not be same type as the input records. A given input pair may map to zero or multiple output pairs. Analyzing and finding the correct mapper instance is important.

3.4.2 Reducer Class

The Reducer class defines the Reduce job in the MapReduce function. It reduces a set of intermediate values that share a key to a smaller set of values. It includes three primary phases Shuffle, Sort, and Reduce. [7]

- Shuffle − The Reducer copies the sorted output.
- Sort −The framework merge-sorts the Reducer inputs by keys. While outputs are being fetched, they are merged so the shuffle and sort is worked simultaneously.
- Reduce −In this phase the value is obtained from the reduce method.

3.4.3 MapReduce Algorithm

1. Client submits the Job to Job Tracker.
2. Job Tracker asks Name Node the location of data.
3. Name Node replies and Job Tracker then saves it in respective Task Tracker.
4. All the results are stored on some database and Name Node is informed.
5. Task Tracker informs Job Tracker completion and progress to Job Tracker.
6. Job Tracker informs client.
7. Client contacts Name Node and gets result.

3.4.4 Map Abstraction

It inputs a key/value pair where key is a reference to the input value and value is the data set on which to operate. It is a function defined by user to apply every value in value input. Produces a new list of key/value pairs they are can be different type from input pair.

```
def map(key, value)
    list =[ ]
    for x in value:
        if test:
            list.append((key,x))
    return list
```

3.4.5 Reduce Abstraction

It starts with intermediate key/value pairs and ends with finalized key/value pairs. Starting pairs are sorted by key Iterator which supplies the values for a given key to the Reduce function.

```
def reduce (key,listOfValues):
    result = 0
    for x in listOfValues:
        result += x
    return (key, result)
```

3.4.6 Example:

Step 1. Input files “Apple Orange Mango Orange Grapes Plum Apple”

Step 2. Individual mapper instance “Apple Orange Mango” “Orange Grapes Plum Apple”

Step 3. Key value to the instance “Apple=1 Orange=1 Mango=1 Orange=1 Grapes=1 Plum=1 Apple=1”

Step 4. Sort and shuffle “Apple=1 Apple=1 Mango=1 Orange=1 Orange=1 Grapes=1 Plum=1”

Step 5. Reduce Key Pair “Apple=2 Mango=1 Orange=2 Grapes=1 Plum=1”

Step 6. Final output “Apple=2 Mango=1 Orange=2 Grapes=1 Plum=1”

IV. CONCLUSION

The availability of Big Data is a low cost commodity and new information management for data analytics. These capabilities are neither theoretical nor trivial as they represent a major leap on the current opportunities and productivity that can be obtained for the analysis of it. Considering all this understanding the data and the technique to handle the data is an essential part of big data. The above paper represents the various analytic tools to perform the data analysis. Each represents a different way and generates a different outcome that is being needed for the further processing.

REFERENCES

[1] S. Anthony , F. M. R. Gilles, A.-K. K. C. Samer and F. Ian , "Active Data: A Programming Model to Manage Data Life Cycle Across Heterogeneous Systems and Infrastructures," IEEE, 2015.

[2] B. P. Aditya , B. Manashvi and N. Ushma , "Addressing big data problem using Hadoop and Map Reduce," IEEE, 2013.

[3] L. Xiaoyi, R. Md. Wasi , I. Nusrat , S. Dipti and K. P. Dhabaleswar , "Accelerating Spark with RDMA for Big Data Processing: Early Experiences," IEEE, 2014.

[4] L. Sarah , N. Dylan , K. YongChul , H. Bill , B. Magdalena and G. Jeffrey P. , "Analyzing Massive Astrophysical Datasets: Can Pig/Hadoop or a Relational DBMS Help?," IEEE, 2009.

[5] T. Ashish , S. S. Joydeep , J. Namit , S. Zheng , C. Prasad , Z. Ning , A. Suresh , L. Hao and M. Raghotham , "Hive – A Petabyte Scale Data Warehouse Using Hadoop," IEEE, 2010.

[6] G. Katarina , H. Michael , H. Wilson A. , L. Alexandra , A. David S. and C. Miriam A.M. , "Challenges for MapReduce in Big Data," IEEE, 2014.

[7] E. Jaliya , P. Shrideep and F. Geoffrey , "MapReduce for Data Intensive Scientific Analyses," IEEE, 2009.