IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Neoteric Study on Big Data

Ishani Wadhwana[1], Manisha Valera[2]

Student, Computer engineering Dept., Indus Institute of Technology and Engineering, Ahmedabad, India[1]

Assistant Professor, Computer engineering Dept., Indus Institute of Technology and Engineering, Ahmedabad, India [2]

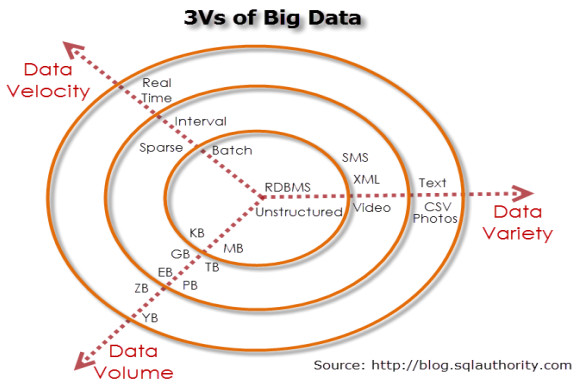
Abstract: The term Big Data has been coined to refer to the gargantuan bulk of data that cannot be dealt with traditional data-handling techniques. Big data is new archetype that has been discovered in past years. We have reached in era of interconnectivity among different organization to predict and make decisions in time changing environments. Big Data is the term for any gathering of datasets so vast and complex that it gets to be distinctly troublesome to process using traditional data processing techniques. The challenges include capture, storage, analysis, data curation, search, transfer, visualization, quering, updating and information privacy. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data. Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, select, manage, and process data within time.

Keywords: BIG DATA, 3V's**,** HDFS, MAP REDUCE, HIVE, PIG, HBASE.

INTRODUCTION

Every day, we create 2.5 quintillion bytes of data so much that 90% of the data in the world today has been created in the last two years alone. This data comes from many sources such as sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few[1]. The term "Big Data" has recently been applied to datasets that grow so large that it gets arduous to work with using traditional database management systems. Non-relational databases, such as Not Only SQL (NoSQL), were developed for storing and managing unstructured, or non-relational, data. NoSQL databases aim for massive scaling, data model flexibility, and simplified application development and deployment. NoSQL databases separate data management and data storage. Such databases rather focus on the high-performance scalable data storage, and allow data management tasks to be written in the application layer instead of having it written in databases specific languages.

3V's OF BIG DATA:



(A) VOLUME:

The first characteristic of Big Data, which is in been discussed is "Volume". Volume is the V most associated with big data because, volume can be huge. What we're talking about here is quantities of data that reach almost unintelligible portions. Facebook, for example, stores photographs. That statement doesn't begin to wonder the mind until you start to realize that Facebook has more users than China has people. Each of those users has stored a whole lot of photographs. Facebook is storing roughly 250 billion images. Data volumes are expected to grow 50 times by 2020[2].

(B) VARIETY:

When talked about sensor data, tweets, photographs, encrypted packets, each of these are very different from each other. This data isn't the old rows and columns and database joins of our forefathers. It's very different from application to application, and much of it is unstructured. That means it doesn't easily fit into fields on a spreadsheet or a database application. Photos, videos, audio record-ings, email messages, documents, books, present-ations, tweets and ECG strips are all data, but they're generally unstructured, and incredibly varied. All that data diversity adds up to the variety vector of big data. Data produced are from different categories, consists of unstructured, standard, semi structured and raw data which are very difficult to be handled by traditional systems. Data today comes in all types of formats. Structured, numeric data in traditional databases. Information created from line-of-business applications. Unstructured text documents, email, video, audio, and financial transactions [7].

(C) VELOCITY:

250 billion images may seem like a lot. But if you want your mind blown, consider this: Facebook users

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

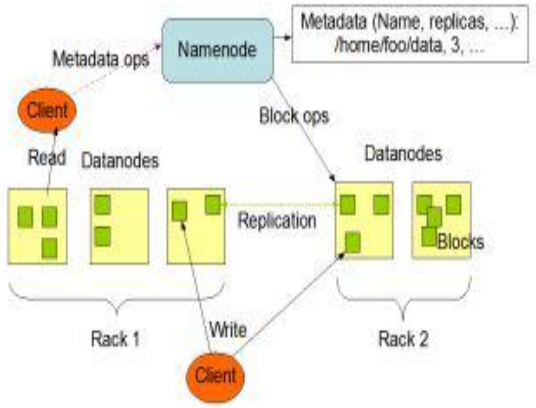
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

upload more than 900 million photos a day. So that 250 billion number from last year will seem like a drop in the bucket in a few months. Velocity is the measure of how fast the data is coming in. Facebook has to handle a great amount of photographs every day. It has to ingest it all, process it, file it, and somehow, later, be able to retrieve it. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time. Reacting quickly enough to deal with data velocity is a challenge for most organizations[8].

TECHNOLOGIES AND METHODS:

Hadoop is a distributed software solution. It is a scalable fault tolerant distributed system for data storage and processing. There are two main components in Hadoop:
(A) HDFS (which is a storage)
(B)Map Reduce (which is retrieval and processing)

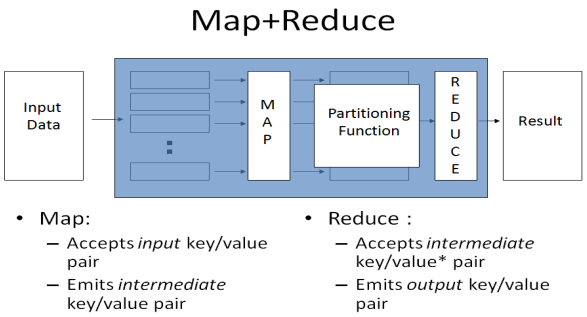
(A)HDFS:
HDFS is highly fault tolerant and it is designed with the help of low-cost hardware. HDFS holds huge amount of data and provides easier access. To store such huge amount of data, the files are stored across multiple machines in redundant fashion fashion to rescue the system from possible data losses in case of failure. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive PCs. In the event that one fizzles, Hadoop keeps on working the group without losing information or hindering work, by shifting work to the remaining machines in the cluster .HDFS also makes applications available to parallel processing. HDFS manages storage on the cluster by breaking approaching records into pieces, called "squares," and putting away each of the squares redundantly across the pool of servers. In the common case, HDFS stores three complete copies of each file by copying each piece to three different servers[3].



(B) MAP REDUCE:
Map Reduce is a parallel programming model, inspired by the "Map" and "Reduce" of functional languages, which is suitable for big data processing. It is the core of Hadoop, and performs the data processing and analytics functions. The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.
MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

Map stage : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

Reduce stage : This stage is the combination of the shuffle stage and the reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.



PIG:
Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with **Hadoop**; we can perform all the data manipulation operations in Hadoop using Apache Pig. To write data analysis programs, Pig provides a high-level language known as Pig Latin. This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data. To analyze data using Apache Pig, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a component known as Pig Engine that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.Let'sfirst look at the programming language itself so that you can see how it's significantly easier than having to write mapper and reducer programs.
LOAD : As is the case with all the Hadoop features, the objects that are being worked on by Hadoop are stored in

IJARCCE

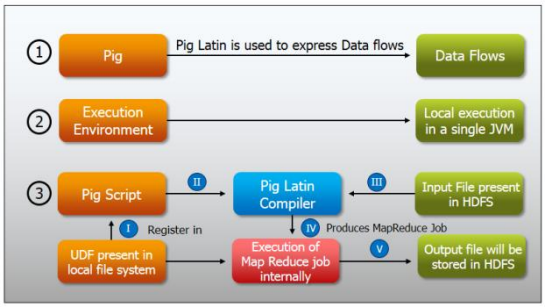
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

HDFS. In order for a Pig program to access this data, the program must first tell Pig what file (or files) it will use, and that's done through the LOAD 'data_file' command (where 'data_file' specifies either an HDFS file or directory). If a directory is specified, all the files in that directory will be loaded into the program. If the data is stored in a file format that is not natively accessible to Pig, you can optionally add the USING function to the LOAD statement to specify a user-defined function that can read in and interpret the data[4].

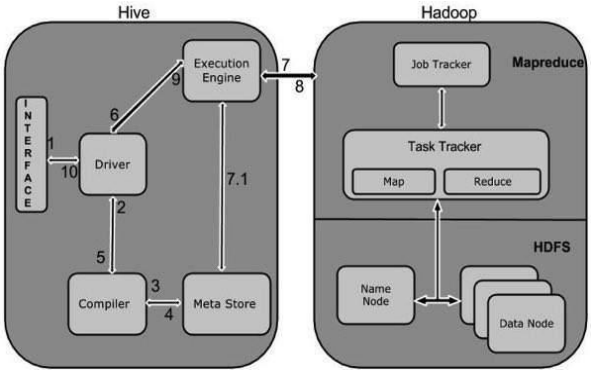
TRANSFORM : The transformation logic is where all the data manipulation happens. Here you can FILTER out rows that are not of interest, JOIN two sets of data files, GROUP data to build aggregations, ORDER results, and much more.

DUMP and STORE: If you don't specify the DUMP or STORE command, the results of a Pig program are not generated. You would typically use the DUMP command, which sends the output to the screen, when you are debugging your Pig programs. When you go into production, you simply change the DUMP call to a STORE call so that any results from running your programs are stored in a file for further processing or analysis. Note that you can use the DUMP command anywhere in your program to dump intermediate result sets to the screen, which is very useful for debugging purposes



Hive:

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy. The following table defines how Hive interacts with Hadoop framework[5]:



1)Execute Query:The Hive interface such as Command Line or Web UI sends query to Driver (any database driver such as JDBC, ODBC, etc.) to execute.

2) Get Plan:The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.

3) Get Metadata:The compiler sends metadata request to Metastore (any database).

4) Send Metadata:Metastore sends metadata as a response to the compiler.

5) Send Plan:The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.

6) Execute Plan:The driver sends the execute plan to the execution engine.

7) Execute Job:Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.

7.1) Metadata Ops:Meanwhile in execution, the execution engine can execute metadata operations with Metastore.

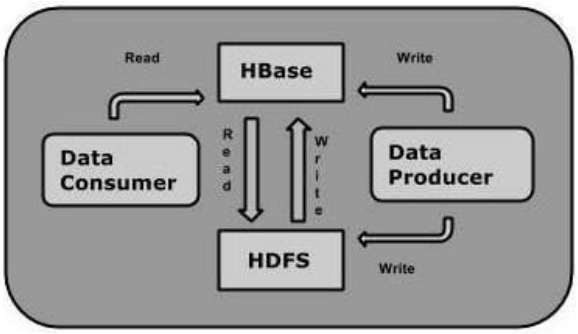
8)Fetch Result:The execution engine receives the results from Data nodes.

9)Send Results:The execution engine sends those resultant values to the driver.

10) Send Results:The driver sends the results to Hive Interfaces.

Hbase: HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.

HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.



USAGE AREAS ON BIG DATA:

Big data is used efficiently in numerous fields. Some of them are listed below[6]:

1) Automotive industry
2) High technology and industry
3) Oil and gas
4) Telecommunication sector
5) Medical field
6) Retail industry
7) Packaged consumer products
8) Media and show business
9) Travel and transport sector
10) Financial services
11) Social media and online services
12) Public services
13) Education and research
14) Health services
15)Law enforcement and defense industry

CONCLUSION

This paper showed us the various technologies to handle large amount of data. Consequently big data was discussed. Data storage management and some analytical tools were examined. By applying such analytics to big data, valuable information can be extracted and exploited to enhance decision making and support informed decisions. Also the usage on big data in our day to day has been given. Thus we have developed a better understanding on big data after we have been able to put words to it.

REFERENCES

[1] Samiddha Mukherjee and Ravi Shaw , Big data-Concepts,application and future scope. Information Technology, Institute of Engineering & Management, Kolkata, India. International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 2, February 2016.
[2] Priyank Jain, Manasi Gyanchandani, Nilay Khare, Dhirendra Pratap Singh, and Lokini Rajesh, A Survey on Big Data privacy using Hadoop Architecture. CSE Department ,MANIT, Bhopal M.P India IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.2, February 2017.
[3] Priyank Jain, Manasi Gyanchandani, Nilay Khare, Dhirendra Pratap Singh and Lokini Rajesh,CSE Department,MANIT, Bhopal M.P India. IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.2, February 2017.
[4] Ms. Vibhavari Chavan and Prof. Rajesh. N. Phursule, Survey Paper On Big Data Department of Computer Engineering, JSPM's Imperial College of Engineering and Research, Pune Vibhavari Chavan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7932-7939.
[5] https://www.tutorialspoint.com/hive
[6] Tanvi Ahlawat and Dr. Radha Krishna Rambola School Of Computing Science and Engineering Galgotias University, Greater Noida, Literature review on big data. International journal of advancement in engineering technology, management and applied science, volume 3, Issue 5.may-2016.
[7] Ms. Vibhavari Chavan, Prof. Rajesh. N. Phursule,‖ Survey Paper On Big Data‖ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7932-7939
[8] Manisha Valera, Ankit Virparia, Om Mehta AN EXHAUSTIVE STUDY: BIG DATA International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5, Issue 6, June 2016