Special Issue of Engineering and Scientific International Journal ISSN 2394-7187(Online) ISSN 2394-7179 (Print)
Proceedings of the UGC Sponsored - National Conference on "Recent Statistical Computing Techniques and their Applications"
Department of Statistics and Department of Computer Science, Ramanujan College,University of Delhi, Delhi.
March 11-12, 2016. *Published on January 2017*

Big Data : Real Challenge

*Dr. Arpita Aggarwal[#1] , Purnima Khurana[*2], Ishan Rathi[*3], Kashish Singh[*4]*
[1]*Associate Professor-PGDAV College, University of Delhi*
[2]*Assistant Professor-PGDAV College, University of Delhi*
[3]*PGDAV College, University of Delhi*
[4]*PGDAV College, University of Delhi*

Abstract— In this paper, we review the background, state-of-the-art and management of big data. Big data is a large volume of structured and unstructured data which is too large to handle using traditional databases. It needs to be analyzed for determining various patterns to make better decisions. Its challenges include storage, analysis, search, transfer, visualization, querying and security of information. We give the general background of big data and related technologies, such as cloud computing, Internet of Things, Hadoop and Spark. After that we focus on Hadoop and Spark. Hadoop is an Open source Java based Framework used for processing large amount of data. It is built on simple programming model called MapReduce. Another framework discussed is Apache Spark, which is designed for faster computation. Spark is not a modified version of Hadoop and is not completely dependent on Hadoop because it has its own processing technique. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application. We discuss some applications of big data. This paper aims to provide a comprehensive overview, big-picture and the challenges faced by big data.

Keywords— Big Data; Hadoop; Spark; Map Reduce; HDFS.

1. Introduction

World we know today is a connected place where large amount of information flows every second. It surrounds us and is a vital part of our life. We access it using different computing devices. With so much information flowing around us we require adequate methods to store, retrieve and modify this information. Big Data is the term used for extremely large collection of data which can be analyzed to reveal some patterns of human behavior and interactions so as to help us make strategic decisions. More than 2 billion people and millions of enterprises living their lives and doing their work online; more than 2 billion of TV hours are streamed from Netflix every month. In 2002, Google received more than 2 million search queries every minute. Today Google receives over 4 million search queries per minute [2].

Nowadays vast amount of data is used every minute. 2.5 million pieces of content is used by facebook users, Nearly 300,000 times tweets are done by twitter users. Approximately 220,000 new photos are posted by Instagram users, 72 hours of new video content are uploaded by youtube users, nearly 50,000 apps are downloaded by apple users, and over 200 million messages are send by email users [3]. This shows that the amount of digital data in the world is growing at an exponential rate. The rate of new data generation is more than the rate of analyzing it. According to Professor Patrick Wolfe, Executive Director of the University College of London's Big Data institute, just about 0.5% of all the data is currently analyzed, and this percentage is shrinking as more data is being collected.

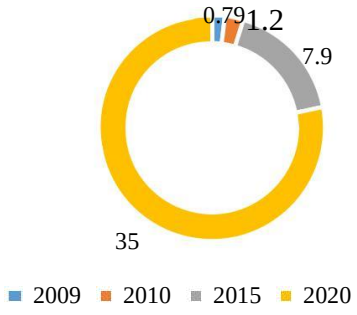

Fig. 1: Growth Patterns of Global data

The potential benefits of Big Data analytics are very significant, and some success has already been achieved but still there remain many possibilities to find better methods to find relevant patterns in Big Data.

The next section of the paper discusses the challenges faced by big data. Section 3 focuses on Hadoop and its tools. Section 4 discusses the properties of Hadoop. Section 5 discusses spark and its benefits.

2. The Four V's of Big Data analysis

The Big data consists of huge amount of data which can be described with four characteristics: volume, velocity, variety and veracity.

Volume— Refers to huge amount of data generated every second. All the images, videos and emails are a part of it. We are talking about Zettabytes of data. This generation of new information is rapidly increasing every year, and this increase makes data sets too large to store and analyze. Big

Data allows us to store this information on multiple clusters and connect these clusters using a network.

Velocity— Refers to speed at which new data is generated and processed. The focus is on getting the information from Big Data in real time with some real time constraint.

Variety— Refers to the different type of information we can store and use now. The data we have analyzed is no longer stored in relation and table. Unstructured Databases are used to store diverse information like images, videos, audio files, digital traces, and sensor data.

Veracity— Refers to the trustworthiness of the data. Accuracy of data is a crucial point. It emphasizes the need for the quality in the Big Data system2.1 Challenges of Big Data

The increase of new data generation is bringing new challenges of data acquisition, storage, management and analysis. Traditional data management systems are based on Relational Databases (RDBMS). Relational database can only be applied to structured data where as in big data heterogeneous data is being generated every second in the form of semi-structured or unstructured data. Structured Database Management Systems are inefficient to handle the huge volume and heterogeneity of Big Data [5].

This huge amount of data poses the challenge of storing this data permanently, managing this data to derive useful information from it. Apart from this, cost of generating and storing this data, elasticity and smooth upgrading and downgrading of data is a big challenge. The Research community has proposed solutions keeping a look at various perspectives. For example, Cloud computing is applied to meet the requirements of cost efficiency, elasticity and smooth upgrading/downgrading. For solutions of permanent storage and management of large tangled datasets, distributed file systems [6] and NoSQL [7] are convenient choices. These frameworks have proved to be proficient in processing clustered tasks of webpage ranking.

The development of big data applications poses big problems [8][9][10]. The key challenges are:-

Data Representation— Data has a variety in type, semantics, organization, accessibility, structure and granularity. Data representation has an aim to make data useable for computer analysis and improper data representation will diminish the value of original data and interrupt productive data analysis.

Redundancy reduction and data compression— In many cases there is high level of redundancy in datasets. This overabundance can increase the cost of entire system and affect the potential values of the datasets. Compression and removal of redundancies reduces the cost of the systems.

Data life cycle management— Sensors and computing systems are generating data at unprecedented rates and scales. One of the obstacles is that current storage systems do not have the capacity to store such enormous datasets. In general terms, value of the big data depends on the novelty of big data. Therefore guidelines are required to decide which data to store and which to reject [5].

Analytical mechanism— Analytics on Big Data is to be done with real time constraints. Traditional Relational database systems lacks the qualities of expansion to meet these real time performance constraints where as Non-relational database systems are proficient in processing of unstructured data and have started to become popular in big data analysis. A lot of research work is still required on the in-memory databases and sample data based on approximate analysis [5].

Expendability and scalability— Today, data creation is exploding at exponential rates. The big data analysis systems must incorporate both present and future datasets.

3. Apache Hadoop

Today when data is being generated in huge volume every minute, the flaws in current database management strategies are coming into the scene such as storage (dynamic/static), handling, security, data retrieval (fastest) and many more implementation and processing specific issues.

The solution comes in the form of Apache Hadoop developed by Douglass Cutting, chief architect at Cloudera(the Apache software Foundation) and Micheal Cafarella, currently computer science professor at University of Michigan[11]. Originally, "Nutch Search Engine Project" urbanized Apache Hadoop. It is used to knob the data produced by the Web Crawlers[11]. Apache Hadoop is scalable, fault- tolerant distributed file system. It has two main systems: "MapReduce" and "Hadoop distributed file system [HDFS]".

The Hadoop's MapReduce is developed using ideas from MapReduce developed by Google for itself and the Hadoop's HDFS is inspired by Google File System. Hadoop also has two more modules apart from MapReduce and HDFS namely [13]:

Hadoop Yarn— Management platform that manages resources in clusters to be scheduled for user's application.

Hadoop Common— Contains all the common files, modules and utilities that other Hadoop modules may need.

3.1 Hadoop Related Projects and Developments

Ambari: A tool operated on web to manage and monitor the Apache Hadoop clusters and the Pig, Hive, Map Reduce Applications and get their performance diagnosed [12].

Avro— To serialize the data.

Cassandra— Multi-master database, scalable with no subsequent failure points [12].

Chukwa— A data collection system for managing large distributed systems.
HBase— Distributed storage for storing database tables as structured storage.
Hive— Provides ad-hoc services and used to summarize the data for Hadoop applications [12].
Mahout— Highly scalable library like Hadoop and used for machine learning and data mining activities.
Pig— Programming language used in Hadoop to support parallel processing.
Spark— It is a major add-on to Hadoop. Its computing power is very high as compared to Hadoop's initial MapReduce. Spark can work stand-alone too on any structured or non-structured database including HDFS. It is applied usually for machine learning and graph computations.
ZooKeeper— A service particularly to be used for coordination purposes in distributed applications.
If achievements are to be talked about then using Hadoop with its add-ons, the following milestones were set in 2009 [12]:

- Fastest sort of a Terabyte in 62 seconds over 1,460 nodes.
- Fastest sort of a Petabyte in 16.25 hours over 3,658 nodes.

The database is stored perfectly on distributed systems and also makes data retrieval and sorting exceptionally faster through Hadoop applications. But with so much of data and data science a new highly profitable application has come to the scene, Recommender systems [12]. Recommender systems are highly useful in "Practical machine learning". The history of searching, browsing, purchases and other data collected through browser or other sources can be converted into something very useful for ROI (Return on Innovation and Investment) to study the user's needs and behavior.

3.2 Hive

Hive [16] is a system that has enabled Hadoop to effectively store, report and analyze data with ease and SQL for querying. This makes Hive a declarative language. It was developed as a translation layer in its early development years. With its SQL like interface, Hive has eased the difficult and stressful programming of MapReduce to analyzethe data stored in HDFS.

The efficiency and working of Hive depends upon the factor how its data warehouse layer is designed, implemented and optimized. Figure 3.1 shows the architecture of Hive. Hive architecture includes several components likely "CLI (Command Line Interface)" and "HiveServer", both used to interact with users and submitting statements to Hive, "Driver" generates Abstract Syntax Tree out of statement, "Planner" to pick precise planner execution to effectively analyze dissimilar parts of AST, "Metastore" stores metadata about the RDBMS, "QueryPlanner" to analyze the queries and convert them to MapReduce jobs, and the "Storage Handler" to process data accumulated in either HDFS or Hbase or else[15]. To start on a service that can access Hive through Server, ODBC or JDBC is used by passing service to the hive command followed by the word "hive server" [14].

If we consider the PigLatin, that specifies the data flow, but in Hive the result is the main focus and then Hive needs to work out best data flow to get that result perfectly. Though like Pig, in Hive to a schema is required, but still we are not limited to only one schema. HiveQL itself is a almost complete language but not a Turing complete language. But still it can be extended through UDFs to be like a Turing complete. Hive working is based on tables. Where there are two program must specify the file (or files) it is going to use. Now for specifying the files to be used there is LOAD 'data_file' command (where 'data_file' specifies either an HDFS file or directory). And if a kinds of tables that can be created: managed tables, where data is managed by Hive and external tables where data is managed outside of Hive. Hive does a thing task called "bucketing" to speed up queries where data is splits up by specific column, but in this bucketing individual values are not specified for the column that corresponds to buckets, we simply have to enter number of buckets and then hive figures out the rest.

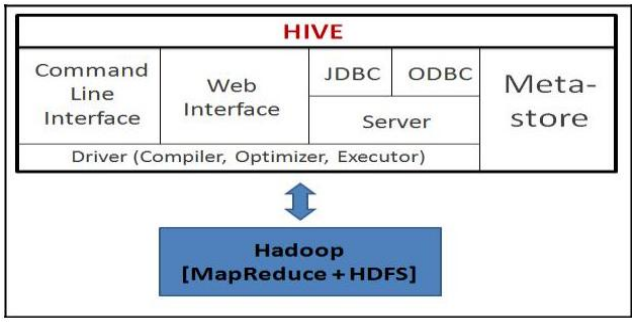

Fig 2: Architecture of Hive

4. Pig

Initially developed at Yahoo!, Pig allows people using Hadoop® to shift there focus on analyzing big data sets rather than spending all the time on writing MapReduce programs. It can handle almost any kind of data. Pig is composed of two components basically: the first is the "PigLatin" language (various Hadoop projects' names have a humorous touch in them), and the second is a runtime environment which executes PigLatin[17] programs. This is such like relationship between a Java Virtual Machine (JVM) and a Java application. Let's have a look at the Programming language to show how easier it is to program

in PigLatin then to keep developing mapper and reducer programs.

- LOAD the data to be manipulated from HDFS.
- Then the data goes through a set of transformations (which, on the backend, is transformed into set of mapper and reduce tasks).
- Finally, either DUMP the data to the screen or else STORE the results in a file.

Load— Considering Hadoop data store HDFS system here. For the Pig program to get access to the data, the directory is specified, all files inside that directory will be loaded into the program. Data stored in any of the file format is not natively accessible to Pig, so we can optionally add the USING function to the LOAD statement to identify a user-defined function which can read and interpret the data.

Transform— This logic specifies all the data manipulation that to be done. Here rows that are not of good interest are FILTER out, JOIN used to join two sets of data files, GROUP data to build aggregations, ORDER results, and much more.

Dump and store— If none out of DUMP and STORE is specified the results of Pig program are not generated. DUMP command, sends the output to the screen, used in debugging Pig programs. For storing the results into a file or set of files we just have to change DUMP command to STORE command and give the information of files where results to be stored for further processing or analysis. As like all other programming languages likely Cpp or Java where output functions are called in between programs to ease debugging process, similarly in Pig DUMP can be used anywhere in between program.

5. MapReduce

Map reduce is a framework for computer clusters to perform distributed processing on large data sets. Map reduce facilitates simple processing of huge datasets in a reliable and fault tolerant way. Map Reduce is an algorithm used in Hadoop, Hadoop cluster integrates MapReduce and HDFS layer, to successfully store, analyze and retrieve the data when needed in shortest time. Actually when it was seen that parallel processing the data stored at different machines is required to be made more efficient and faster then MapReduce programming model was introduced that was very easy to use because it hides the details of storage, parallelization, fault-tolerance and other concepts implemented to make data retrieval safe and efficient. Though simple Map and Reduce functions basically computes intermediate key/value pairs from stored data and then reduce function reduces all the key/value pairs with same key and reduces them to make the required result. Although writing Map and Reduce functions and specifying no. of clusters is almost enough to make data retrieval and data handling easier. But there are some extensions to it:

Partitioning Function: This function is to be given by the user to specify no. of output files and function to split the output generated by the MapReduce into prescribed files in a well-balanced way.

Combiner Function: Directly writing the output files generated through MapReduce to output file can result in significant duplicate data. Specifying combiner functions generate an intermediate output file that is intern send to Reduce function to process.

Side-Effects: Many a times, programmers make auxiliary files with several outputs and then it needs support from application to process the file and make it the final output file through making it atomic and finally generating it. But here we don't specify any application program but else for those times where tasks make multiple output files have to do it with consistency and the files should be deterministic.

Counters: It is basically used to count occurrences of various events where the user has to handle the increment anddecrement of counters inside Map and Reduce functions.

Limitations with Hadoop MapReduceThe major setbacks include inefficiency of MapReduce in running iterative algorithms. MapReduce was basically not designed to handle iterative processes. Mappers read the same data no. of times from the disk where it should write the results back to the disk after each iteration and before next iteration. This highly degrades the performance of MapReduce programs. To handle this, for each iteration, a new mapper and reducer need to be initialized. That proves to be big overhead for the short-lived MapReduce programs. Although some improvements likely to do forward scheduling and setting new MapReduce task before the previous one finishes but this approach to add additional complexity into the source code. One such work called HaLoop that extended the MapReduce with programming support for iterative algorithms and improved the efficiency by adding additional caching mechanisms. CGL MapReduce is another work that focuses on improving the performance of MapReduce iterative tasks

6. Apache Spark

Apache Spark is a framework for performing general data analytics on distributed computing cluster that can run on Hadoop HDFS file system it uses in memory Map Reduce operations.

Apache Spark started out as research project at UC Berkeley in the AMPLab, which focuses on big data analytics. MapReduce is inefficient in handling particular analytical applications, theses application include utilities like: Iterative algorithms, includes many machine learning alrothims.

Interactive data mining, where a user would like to load data in RAM across a cluster. Streaming applications that maintain aggregate state over time.

Spark uses Resilient distributed datasets (RDDs) to handle these applications. RDDs can be stored in memory between queries. RDD allow spark to outperform existing models up to 100x in multi-pass analytics.

6.1 Runs 100 Times Faster than Hadoop MapReduce in Memory

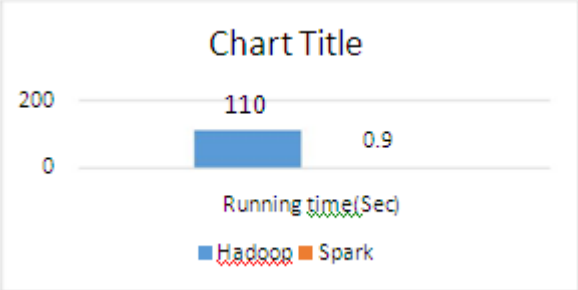

Fig.2: Running time of Hadoop MapReduce

Applications in Java, Python, Scala.

Example: word count in Spark's python API
Text_file= spark.textfile("HELLO.txt")
Text_file.flatmap(lambda line: line.split())
.map(lambda word: (word,l))
.reduceByKey(lambda a, b: a+b)

6.2 Spark runs on Hadoop, Mesos, Standlone, or in the Cloud

Since its release it used across wide range of industries like Yahoo,Baidu and Tencent. These companies have used Spark on massive scale, processing multiple Perabytes of data on clusters of over 8000 nodes. It has quickly become the largest open source community in big data, with over 750 contributors from 200+ organizations.

7. Big Data Applications and Potentials

Big data aims to enhance decision making for companies and Public administrations which will create a notable growth in world economy. From the report of McKinsey institute [18], big data helps to listen to costumers, understands their ways of using services. [19] The applications of big data have profits in many fields from National security to global economy and society administration.

Some of the applications of Big Data are,

Smart cities— Big data brings new possibilities to transform cities and countries. Major objectives of these smart cities and countries are sustainable economic development, high quality of life with wise management of natural resources [19].

Health— Medicines and public health are the field in which data generated at unprecedented rates if analyzed can improve the quality of treatment as well as reduce care cost. The computing power of big data allow us to analyze and store entire DNA strings which can help us to monitor health situation before they become a drastic threat to someone's wellbeing.

Business and Commerce— Over the past 20 years, data has increased on unprecedented rates ,the overall copied and created data globally in various fields was around 1.8 ZB which almost increased to nine times in the last five years [20] this generation of data will double at least every two years in the coming future.

The statistics hidden inside the mass of data can help to optimize business operations, understand customer requirements and needs.

8. Conclusion

Big-Data Problem: It is quite evident that the big data problem is very efficiently solved for current time because now the data is being processed and analyzed at greater speeds with improved Mapping and Reducing Algorithms using HADOOP. But as the data is being generated in Peta Bytes now daily it still needs critical improvements in all fields such as storage, processing, accessing, analyzing and security. That's where the Apache Spark comes in that makes use of highly dynamic Python, and other lightning fast algorithms to achieve 100 times greater speed than Hadoop but still parallel data processing and ETL tasks are still strong points in favor of MapReduce. So for getting optimized and data mapping faster through Spark it either needs to be evolved and innovated for parallel processing or is made to be used alongside Hadoop MapReduce to provide parallel processing or else Hadoop MapReduce can be innovated on very core levels to make its mapping and reducing faster.

The paper provides an overview of both the frameworks. Our observation has showed spark to be the way ahead in Big Data Analysis. Spark is gaining popularity at many places where previously Hadoop MapReduce was in use because it's faster and simple to use.

References

[1] http://www.csc.com/insights / flxwd / 78931 - big _ data_universe_ beginning_to_explode
[2] http:// www. emc. com /leadership /digital - universe / 2014 iview /executive -summary.htm
[3] http://aci.info/2014/07/ 12/the- data-explosion-in- 2014-minute-by-minute-infographic/
[4] http:// www.businessinsider.in/ The-mind- blowing- growth-and-

power-of-big-data/articleshow/47606515.cms
[5] Min Chen•Shiwen Mao•Yunhao Liu Big Data: A Survey. Springer Science and Business Media New York 2014
[6] Howard JH, Kazar ML, Menees SG, Nichols DA, Satyanarayanan M, Sidebotham RN, West MJ (1988) Scale and performance in a distributed file system. ACM Trans Comput Syst (TOCS) 6(1):51–81
[7] Cattell R (2011) Scalable sql and nosql data stores. ACM SIGMOD Record 39(4):12–27
[8] Labrinidis A, Jagadish HV (2012) Challenges and opportunities with big data. Proc VLDB Endowment 5(12):2032–2033
[9] Chaudhuri S, Dayal U, Narasayya V (2011),An overview of business intelligence technology. Commun ACM 54(8): 88–98
[10] Agrawal D, Bernstein P, Bertino E, Davidson S, Dayal U,Franklin M, Gehrke J, Haas L, Halevy A, Han Jetal "Challenges and opportunities with Big Data. A community whitepaper developed by leading researches across the United States (2012)
[11] https://en.wikipedia.org/ wiki/Apache_Hadoop
[12] http:// readwrite. com/ 2010/ 12/06 /top-10 -enterprise- products-of-2010
[13] http://hadoop.apache.org
[14] S. Dhawan et al., American International Journal of Research in Science, Technology, Engineering & Mathematics, 2(1), March-May, 2013, pp. 88-93
[15] Hive by Yin Huai, Ashutosh Chauhan, Alan Gates, Gunther Hagleitner, Eric N. Hanson,Owen O'Malley, Jitendra Pandey, Yuan Yuan, Rubao Lee, Xiaodong Zhang, The Ohio State University : "Major Technical Advancements in Apache"
[16] Apache Hive. Available at http://hive.apache.org
[17] Apache Pig. Available at http://pig.apache.org
[18] C.L. Philip Chen, and Chun-Yang Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data", Informatics and Computer Science Intelligent Systems Applications, Volume 275, 2014
[19] Fatima EL Jamiy, Abderrahmane Daif , Mohamed Azouazi and Abdelaziz Marzak Hassan II University, Faculty Of Sciences Ben m' Sik, Laboratoire Mathématiques Informatique et Traitement de l'Information MITI, Casablanca, Morocco :"The potential and challenges of Big data - Recommendation systems next level application"
[20] Gantz J, Reinsel D (2011) Extracting value from chaos. IDC iView, pp 1–12