Big Data Scrutiny and Evaluation of Hadoop

Ahmed Hussain
Department of Computer Science & Engineering
ASCT, Bhopal, India

Abstract— Utilization of internet, convivial networking sites, astute phones, censor contrivances, monitoring contrivances, online shopping, transactions in stock exchanges, individual medical records in hospitals, images and censor data sent by satellites and many of such kind are swelling at a very high rate day by day despite drop of cost of storage contrivance. Consequently the World is accumulated with astronomically immense amount of data of different kind and structure at each moment. Traditional Data Base Management System or Distributed Database Management System or Structured query language does not have enough flexibility to store and analysis these immensely colossal volumes of ever growing multi-structured data. However these data sparsely contain much consequential information with considerable business values for ecumenical economy and additionally contain information or statistics to be utilized for gregarious welfare schemes. This paper addresses research issues with massively colossal Data Analysis, emergence of Immensely Colossal Data analysis technologies from last decade along with its draw backs and further amelioration scope, and additionally addresses impact of immensely colossal Data analysis on individual and society.

Key words: Big Data, Hadoop, NoSQL, HDFS, MapReduce, CAP Theorem

I. INTRODUCTION

After each text message or photo being posted in social After each date in the form of text message, photo, audio, video being posted in convivial networking sites like Facebook or Twitter or LinkedIn data get engendered for accumulation. Astute phone, which is very prevalent now a day, engenders data by sending GPS signal frequently for ecumenical situating. Shopping done utilizing a credit or debit cards engenders data which is utilizable for targeting customer. Virtually eighty percent of digital data today subsist ecumenical is being engendered in recent four to five years. The data thus engendered is expanding exponentially. These data are intricate because it consists of structured data like bank transaction, unstructured data like text message conversations or video streaming. These types of data are called sizably voluminous Data. Immensely Colossal data can be identified by six main characteristics. These characteristics are Volume, Velocity, Variety, Veracity, Value and Visualization. Astronomically immense data are generally of sizably voluminous volume, growing perpetually at each moment and may consist of data of different format. Data should be engendered from some genuine sources and it is not some junk data engendered by a malware. Moreover Astronomically Immense Data should have potential to engender consequential information and can be visualized. Some other mundane sources of astronomically immense Data generation are e-commerce business, convivial media and different kind of internet applications, sensor networks, stock exchange transactions etc.

A. Big Data Research Challenges

Main objective of astronomically immense Data analysis is to engender value from immensely colossal amount of unorganized data. Privacy and security issues, data ownership, heterogeneity, timeliness, maintaining cloud Accommodation for Sizably Voluminous Data, machine learning algorithm for Astronomically immense Data, scalability and involution are the major research challenges for Immensely Colossal Data analysis. Due to high rate of data magnification, due to sizably voluminous volume and unstructured nature of data, traditional RDBMS and SQL can't be utilized for sizably voluminous Data analysis. To store expeditious growing sizably voluminous information engendering from sundry data source, the data processing system should have a scalable architecture. Scalability designates facility to integrate more node to the cluster as the data grow, without affecting the performance of the system. Traditional RDBMS is not felicitous enough for the Sizably Voluminous Data. First reason is RDBMS or traditional Distributed Data Base System cannot expand to a cluster having thousands of nodes due to restrictions imposed by ACID constraints. In case of cluster with astronomically immense number of nodes there involves consequential network delay and maintaining consistency becomes very arduous. Second reason is traditional RDBMS cannot operate on unstructured or semi structured data. A good sizably voluminous Data analysis system should have two characteristics. Firstly, it should able to store and access immensely colossal volume of data in a minuscule time. Though the storage contrivances becoming more frugal Day by day, the data access haste is not ameliorating in that way. So the data storage architecture should be perspicacious enough to access astronomically immense data in minute time from many slow contrivances. Google Distributed File system and Hadoop Distributed File System are two very efficient frameworks for storage and access of astronomically immense data. Second characteristics of Astronomically Immense Data analysis system is it should be able to process immensely colossal amount of data in diminutive time to draw some conclusion from it. But there is a constraint in microprocessor celerity. Processor speed cannot be incremented beyond certain limit due generation of uncontrollable heat. Ergo parallel data processing is an Alternative solution for data intensive operation. Map Reduce is an innovative conception for data intensive computation which ultimately does parallel processing of sizably voluminous data Google is key player and major contributor towards immensely colossal data analysis technologies. Google publishes three white papers to address the issue of immensely colossal Data storing and processing technique during the period of 2003 to 2006. They are namely – "The Google File System", "Map Reduce: Simplified Data Processing on Astronomically immense Clusters", and "Big table: A Distributed Storage System for Structured Data". These three white papers have paramount effect on the

magnification of Sizably Voluminous Data processing technologies. They magnetized consequential attention from database and parallel computing research community as well as from corporate world. As a result of which Hadoop Distributed File System, Hadoop Map Reduce and some NoSQL data base systems like HBase, Cassandra, MongoDB and a few more come into subsistence and the process is perpetuated till date. Currently they are playing paramount roles in gregarious networking, advertise targeting, ecommerce, data analysis and data management industry and Partition toleration (AP) or Consistency and Partition Tolerance (CP), i.e. only upon a single edge of the triangle. Cassandra, MongoDB and HBase are examples of three very popular NoSQL data base. Cassandra is a highly scalable NoSQL data base predicated on eventual consistency. So it can be placed by the AP side of CAP theorem triangle in Fig. 1. HBase is another popular NoSQL data predicate which is built on top of a sizably voluminous Data processing framework called Hadoop. In HBase predilection is given towards Consistency and Scalability where Availability is undermined. So it can be placed by the CP side of CAP theorem triangle in Fig.1. However in design of MongoDB more predilection is given towards Consistency (C) and it is not tolerant to partition. So MongoDB can be placed by the CA side of the CAP theorem in Fig.1. If we have to keep RDBMS in Fig.1, though it is not a NoSQL data base, we can keep it by CA side of the triangle.

B. Not Only SQL DB

NoSQL originally referring to "non SQL", "non-relational" or "not only SQL predicated RDBMS. To analysis data which cannot be stored in a predefined fine-tuned schema, in such cases NoSQL data base is utilizable. More over for immensely colossal scale data management where traditional RDBMS cannot scale well along with maintaining rigorously the ACID constraints, NoSQL data base is a better supersession in such cases. In lieu of stringent consistency, NoSQL is predicated on eventual consistency.

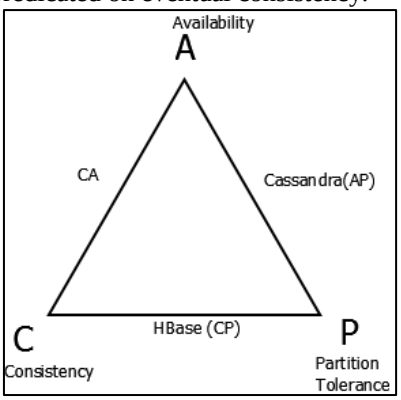

Fig. 1: CAP Theorem

Eventual consistency designates, NoSQL data base ascertains that the data surmise a consistent state at some future point in time. The rudimentary principle of NoSQL data base is CAP theorem. Edifier Eric Brewer put forward the famous CAP theorem in the year 2000. The three consequential requisites of a distributed data base system are- Consistency (C), Availability (A) and Partition Tolerance (P). CAP theorem states that an astronomically immense scale distributed system cannot meet simultaneously all the three requisites, but can only meet two of them at a time. CAP theorem is represented in Fig. 1 as a triangle where each point is one of the three requisites of C, A and P. According to CAP theorem the design of a NoSQL database can be stressed upon either Consistency and Availability (CA) or Availability

C. Hadoop Distributed File System for Big Data Storage

Hadoop Distributed File System, popularly kenned as HDFS is a highly scalable distributed file system with high availability. It is an open source project under Apache Hadoop Software Substructure. Main advantages of HDFS are, it is highly fault tolerant and it can be run on thousands of low cost commodity machines for data intensive computation. Being a scalable Architecture it is felicitous for astronomically immense Data storage and access. A Hadoop cluster consists of mainly two kinds of nodes, a single name node and thousands of data nodes. Name node is a reliable machine with high configuration which stores all metadata about the whole file system in the cluster. Authentic data are stored across an astronomically immense number of data nodes. Data nodes are commodity machines with low cost. To eschew data lose same data are stored in multiple replicas across data nodes in the cluster. If one data node fails, replicas of data present in that machine are still available on some other machines. To read or write data into a HDFS cluster, a client have to first communicate with name node to access the Meta data. Name node is often called master node and data node are called slave nodes. Data nodes periodically send heart bit signals to denominate node to denote that they are functioning opportunely.

The Apache Hadoop HDFS project is incentivized by the white paper published by Google describing its distributed file system called GFS or Google File system. If sizably voluminous amount of data are stored in a single machine with high storage capacity and circumscribed input-output channel, it takes more time to write and read data from that single machine. Parallel data access is not possible in such cases. In lieu of that, if immensely colossal scale data are stored across multiple machines, data can be accessed in parallel by taking lesser time. Main principle of GFS is storing and accessing astronomically immense scale data in parallel in a lesser time. Apache Hadoop additionally implements the same principle.

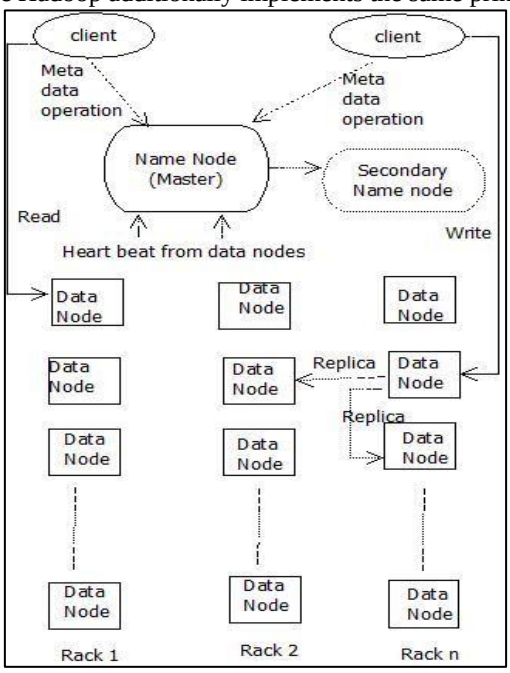

Fig. 2: HDFS: Hadoop Distributed File System

Since data nodes are low cost machines, chances of their failures are high. To surmount the quandary of data lose, a configurable replication factor is utilized. The number of replicas maintain for a data block is equipollent to the replication factor, which can be configured verily as per the consequentiality of data. Even if all nodes of a rack fail, still data remain safe in another rack due to rack cognizant replication policy. HDFS maintains at least one replica of a data in a different rack. That's the reason of highly fault tolerance of HDFS.

Conventionally a general file system maintains minutely minuscular block size, which may be only of a few Kilobytes. However it is different in case of HDFS. HDFS block size is much more immensely colossal as compared to other file system. Data in HDFS is stored as a minimum block size of 64 MB or even more. One reason for more astronomically immense block size is to minimize the size of metadata. Storage and mapping information of each block is stored as metadata in the denomination node. To access a file a client has to get the metadata of that file from the designation node. After fetching the metadata of a file, it comes to ken about all the data nodes where the file blocks are stored across. As the block size increases number of blocks required to store a file decreases. As a result the overall metadata size additionally decreases. Consequently it becomes more facile to manage the metadata and it avails in incrementing performance of a Hadoop cluster. Another reason for more immensely colossal block size in Hadoop is to minimize the wastes of seek time. For an astronomically immense data block the time required to transfer the data from disk is significantly longer than the seek time to locate the commencement of the block. This benefits over seek time is not possible in case of immensely colossal number of blocks with minute size.

D. Map Reduce Data Processing

MapReduce is a parallel data processing technique and it accentuated on computation predicated on data locality. MapReduce fits well for processing mass volume of data stored across astronomically immense number of nodes in a cluster homogeneous to HDFS. The main conception of MapReduce is to bring computations to the data nodes where the involved data are present, in lieu of bringing data to some nodes which are yare for computation. Obviating dispensable data movement by computing data at local nodes preserves a plethora of network band width and time. This is a key factor for better efficiency of Hadoop. The role of MapReduce computation and HDFS storage are performed by the slave nodes in a HDFS cluster.

The tasks performed in slave nodes are of two kinds, map tasks and reduce tasks. All possible parallel computations are done by map tasks and outputs of map tasks are determinately processed by Reduce tasks. MapReduce frame work customarily split the input data into some independent data chunks which can be processed in parallel. Those data chunks are processed in parallel by map tasks. The framework sorts the output of map tasks and redirect them to the opportune reduce tasks. Fig. 3 shows the data flow in a simple map reduce computation.

E. Drawbacks and improve scope

Although Hadoop is one of the popular and famous sizably voluminous Data analysis frame work, it is not liberate from pitfalls from its very beginning. Many enhancements are being made since its beginning and the process is still going on. All nodes of a Hadoop cluster is monitored and managed by a Master node. Authentic data are stored in a file system across thousands of data nodes and a Master node keeps track of the whole file system metadata. If Master node fails the whole cluster become in accessible. Ergo master node is a single point of failure in earlier relinquishment of Hadoop. Though Master node is a high configuration machine, still its probability of failure cannot be eliminated. To surmount single point of failure, a secondary name node is introduced. It only keeps duplicate replica of metadata, and maintains check points of different operations at different point of time. In case of failure of name node, metadata from secondary name node required to be facsimiled. Thus the instauration process takes some time. So long instauration time is a major quandary in this approach. In later relinquishment of Hadoop, remedies are done by maintaining pair of name nodes, one in active mode and another in standby mode. In event of failure of active name node, the stand by name node surmounts its obligations in a diminutively minuscule interval of time. There involves no delay of coping Metadata as both the denomination nodes keeps updated metadata. However to maintain two name nodes there incur some superfluous overheads like updating metadata to both the designation node besides the cost incurred in purchasing two high configuration machines.

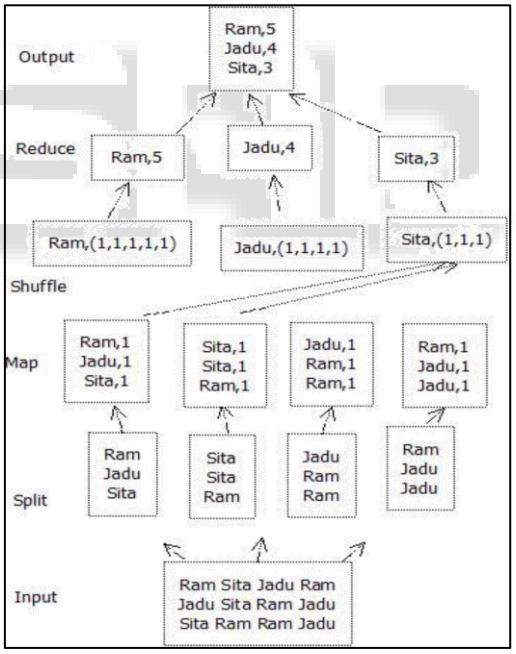

Fig. 3: Hadoop-MapReduce

In Hadoop when a utilizer submits a Job it is consummated by coordination of a single Job Tracker process and multiple Task Tracker processes. Task Trackers are the processes which run on thousands of data nodes and perform the genuine computations. When a utilizer submits a job, a Job Tracker which runs on name node distributes the Jobs to many Task Trackers as minute tasks. Moreover Job Tracker has to monitor all Task Trackers on data nodes and perform resource management along with job scheduling. Thus the single Job Tracker is overloaded with lots of responsibilities. So if number of data nodes keeps on incrementing the performance reduces drastically after a certain limit of approximately 4000 nodes. To surmount this constraint there

should some vicissitudes in Job Tracker and Task Tracker functionality. Responsibility of Job Tracker should be reduced or shifted to other places. So in later version of Hadoop an surplus framework called YARN is introduced for job scheduling and cluster resource management. YARN stands for- "Yet another Resource Negotiator". YARN enhances scalability of Hadoop beyond 4000 nodes by splitting responsibilities of Job Tracker into separate entities. Moreover YARN enhances flexibility of Hadoop to fortify more computation like graph processing besides MapReduce.

F. Big Data Impact on Societies

Emergence of Immensely huge Data technologies made it possible for a wide range of people including researchers from convivial science and humanities, edifying institute, regime organization, and individual to engender, share, organize and interact with sizably voluminous scale data. With what motive and perspective do people from different groups use mass volume of data utilizing latest technology is crucial. If it is utilized for decision making or opinion making or enforcement of incipient policies, it will have considerable long term impact on society and individual. The market visually perceives Astronomically Immense Data as pristine opportunity to target advertising towards right kind of people, which may bother an individual with flood of advertisements. Business and regimes may exploit sizably voluminous Data without concern for issue of legitimacy, data quality. This may leads to poor decision makings. The threat of avail of Immensely Colossal Data without a licit structure and rigorous law can hamper both individual and society holistically. Sizably voluminous data does not always mean as better data. A few Convivial scientists and policy maker optically discerns sizably voluminous data as a representative of society. Which is not obligatorily be true as an immensely colossal portion of world population still does not dump data into Sizably Voluminous Data repository by utilizing internet or by any other betokens. For instance Twitter or Facebook does not represent all people, all though many Sociology researchers and journalist treat them as if they are representative of ecumenical population. Moreover number of accounts on convivial networking sites does not obligatorily represent same number of people, as individuals can feign their identity and can engender multiple accounts. A sizably voluminous mass of raw information in form of immensely colossal Data is not self-explanatory. And the concrete methodologies for interpreting the data are open to all sorts of philosophical and ethnical debate. It may or may not represent the truth and an interpretation may be inequitable by some ethnic views or personal opinions. Personal data can be sensitive and may have some privacy issue. It is valid and earnest issue whether privacy can be maintained with incrementing storage and usages of Astronomically Immense Data. For example there are astronomically immense data on health care system available today which can are utilized extensively for research purport. And an individual can be identified from it and can be monitored periodically who is suffering from a disease without his or her erudition. But it may emotionally or convivially harm the person once his or her health information made public by people with evil intention. Many dataset contains identifier for individual such as denomination, date of inception or unique code issued by regime agencies. So an individual can be spied with good or lamentable intention. Sizably voluminous data aggregator postulates that they have rights to the whole data which may include private and sensitive data as well. But in case of company failure or company surmount, the data set may go to some other hand and any subsisting privacy bulwark policy are unlikely to survive in a hand of an incipient owner

II. CONCLUSION

A technology is not good or deplorable in itself. To utilize it for the welfare of society, astronomically immense Data operator must be gainsaid a free ride by enforcing stringent law and privacy policy to obviate misuse of data for erroneous intention. The Astronomically Immense Data technology is incipient research area and being developing from last decade and there are scopes for amendments. It is playing a consequential role for ecumenical economy, scientific research, enforcement of gregarious welfare scheme, and malefaction detection. Consequentiality of Hadoop, NoSQL data base is incrementing as RDBMS and SQL are not congruous to handle unstructured authentic time data.

REFERENCES

[1] Tom White, "Hadoop the Definitive Guide, 3rd Edition", O'REILLY, 2012.
[2] Apache Software Foundation, Official apache hadoop website, http://hadoop.apache.org/,Oct, 2013.
[3] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System," Google, 2003.
[4] Jeffrey dean, and Sanjay Ghemawat, "MapReduce: Simplified Data processing on Large Clusters," Google, 2004.
[5] Gouxi Wang, and Jianfeng Tang, "The NoSQL Principles and Basic Application of Cassandra Mode," International Conference on Computer Science and Service System, 2012.
[6] Kala Karun. A, and Chitharanjan. K, "A Review on Hadoop - HDFS Infrastructure Extension," 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
[7] Marcus R. Wigan, and Roger Clarke, "Big Data's Big Unintended Consequences," Published by the IEEE Computer Society, 2013
[8] D. Boyd and K. Crawford, "Six Provocations for Big Data," Dynamics of the Internet and Society, Oxford InternetInst.,Sept.2011; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1926431

