INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

BIG DATA ANALYSIS ISSUES AND EVOLUTION OF HADOOP

SURAJIT DAS, DR. DINESH GOPALANI

Department of Computer Engineering, Malaviya National Institute of Technology, Jaipur, India

Accepted Date: 15/02/2014 ; Published Date: 01/04/2014

Abstract: The cost of storage device is decreasing and uses of internet, social networking sites, smart phones, censor devices, monitoring devices, online shopping, transactions in stock exchanges, individual medical records in hospitals, images and censor data sent by satellites and many of such kind are increasing at a very high rate day by day. Therefore the World is flooded with huge amount of data of different kind and structure at each moment. Traditional Database Management System or Distributed Database Management System or SQL does not have enough flexibility to store and analysis these huge volumes of ever growing multi-structured data. However these data sparsely contain much important information with considerable business values for global economy and also contain information or statistics to be used for social welfare schemes. This paper addresses research issues with Big Data Analysis, emergence of Big Data analysis technologies from last decade along with its draw backs and further improvement scope, and also addresses impact of Big Data analysis on individual and society.

Keywords: Big Data; Hadoop; NoSQL; HDFS; MapReduce; CAP Theorem

Corresponding Author: Mr. SURAJIT DAS

Access Online On:

www.ijpret.com

How to Cite This Article:

Surjit Das, IJPRET, 2014; Volume 2 (8): 152-161

PAPER-QR CODE

INTRODUCTION

After each text message or photo being posted in social networking sites like Facebook or Twitter or LinkedIn data get generated for accumulation. Smart phone, which is very common now a day, generates data by sending GPS signal frequently for global positioning. Shopping done using a credit or debit cards generates data which is useful for targeting customer. Almost eighty percent of digital data today exist worldwide is being generated in recent four to five years. The data thus created is expanding exponentially. These data are complex because it consists of structured data like bank transaction, unstructured data like text message conversations or video streaming. These types of data are called Big Data. Big data can be identified by six main characteristics. These characteristics are Volume, Velocity, Variety, Veracity, Value and Visualization. Big data are generally of huge volume, growing continuously at each moment and may consist of data of different format. Data should be generated from some genuine sources and it is not some junk data generated by a malware. More over Big Data should have potential to generate meaningful information and can be visualized. Some other common sources of Big Data generation are e-commerce business, social media and different kind of internet applications, sensor networks, stock exchange transactions etc.

Big Data Research Challenges

Main objective of Big Data analysis is to generate value from huge amount of unorganized data. Privacy and security issues, data ownership, heterogeneity, timeliness, maintaining cloud service for Big Data, machine learning algorithm for Big Data, scalability and complexity are the major research challenges for Big Data analysis. Due to high rate of data growth, due to huge volume and unstructured nature of data, traditional RDBMS and SQL can't be used for Big Data analysis.

To store fast growing huge information generating from various data source, the data processing system should have a scalable architecture. Scalability means ability to add more node to the cluster as the data grow, without affecting the performance of the system. Traditional RDBMS is not suitable enough for the Big 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 large number of nodes there involves significant network delay and maintaining consistency becomes very difficult. Second reason is traditional RDBMS cannot operate on unstructured or semi structured data.

A good Big Data analysis system should have two characteristics. Firstly, it should able to store and access huge volume of data in a small time. Though the storage devices becoming cheaper day by day, the data access speed is not improving in that way. So the data storage architecture should be smart enough to access huge data in small time from many slow devices. Google Distributed File system and Hadoop Distributed File System are two very efficient frameworks for storage and access of huge data. Second characteristics of Big Data analysis system is it should be able to process huge amount of data in small time to draw some conclusion from it. But there is a limitation in micro processor speed. Processor speed cannot be increased beyond certain limit due generation of uncontrollable heat. Therefore parallel data processing is an alternative solution for data intensive operation. Map Reduce is an innovative idea for data intensive computation which ultimately does parallel processing of huge data.

Google is key player and major contributor towards big data analysis technologies. Google publishes three white papers to address the issue of Big 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 Large Clusters”, and “Big table: A Distributed Storage System for Structured Data”. These three white papers have significant effect on the growth of Big Data processing technologies. They attracted significant 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 existence and the process is continued till date. Currently they are playing important roles in social networking, advertise targeting, e-commerce, data analysis and data management industry.

NoSQl Data Base

NoSQL stands for Not Only SQL based RDBMS. To analysis data which cannot be stored in a predefined fixed schema, in such cases NoSQL data base is useful. More over for large scale data management where traditional RDBMS cannot scale well along with maintaining strictly the ACID constraints, NoSQL data base is a better replacement in such cases. Instead of strict consistency, NoSQL is based on eventual consistency. Eventual consistency means, NoSQL data base ensures that the data assume a consistent state at some future point in time. The basic principle of NoSQL data base is CAP theorem. Professor Eric Brewer put forward the famous CAP theorem in the year 2000. The three important requirements of a distributed data base system are- Consistency (C), Availability (A) and Partition Tolerance (P). CAP theorem states that a large scale distributed system cannot meet simultaneously all the three requirements, 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 requirements 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 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 based 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 base which is built on top of a Big Data processing framework called Hadoop. In HBase preference 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 preference 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.

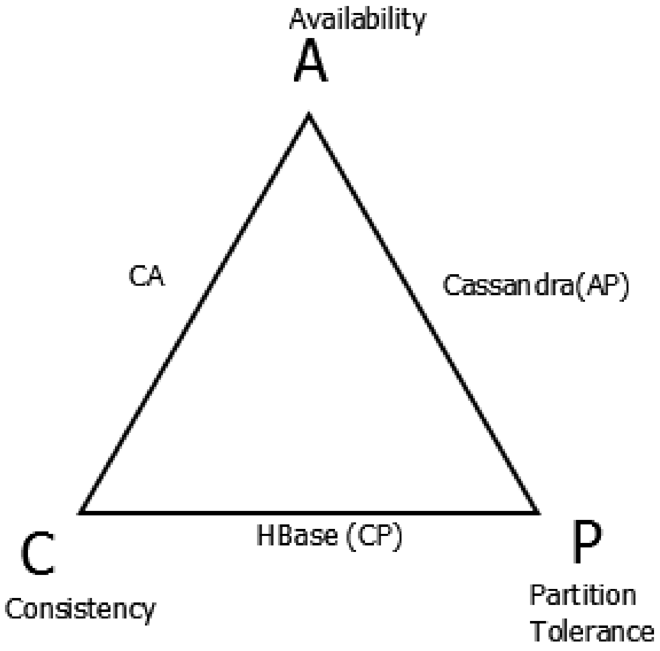

Fig. 1 CAP Theorem

Hdfs for Big Data Storage

Hadoop Distributed File System, popularly known as HDFS is a highly scalable distributed file system with high availability. It is an open source project under Apache Hadoop Software Foundation. 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 suitable for Big 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. Actual data are stored across a large number of data nodes. Data nodes are commodity machines with low cost. To avoid 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 name node to indicate that they are functioning properly.

The Apache Hadoop HDFS project is motivated by the white paper published by Google describing its distributed file system called GFS or Google File system. If huge amount of data are stored in a single machine with high storage capacity and limited 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. Instead of that, if large 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 large scale data in parallel in a lesser time. Apache Hahoop also implements the same principle.

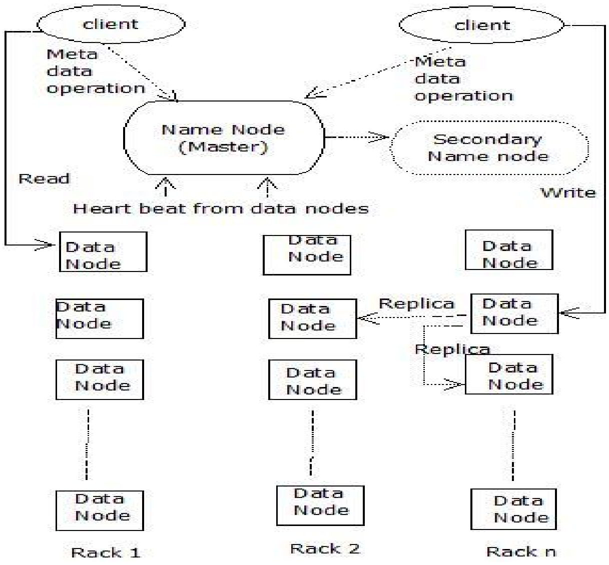

Fig. 2 HDFS: Hadoop Distributed File System

Since data nodes are low cost machines, chances of their failures are high. To overcome the problem of data lose, a configurable replication factor is used. The number of replicas maintain for a data block is equal to the replication factor, which can be configured verily as per the importance of data. Even if all nodes of a rack fail, still data remain safe in another rack due to rack aware replication policy. HDFS maintains at least one replica of a data in a different rack. That is why HDFS is highly fault tolerant.

Usually a general file system maintains very small block size, which may be only of a few kilobytes. However it is different in case of HDFS. HDFS block size is much larger 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 larger block size is to minimize the size of metadata. Storage and mapping information of each block is stored as metadata in the name node. To access a file a client has to get the metadata of that file from the name node. After fetching the metadata of a file, it comes to know 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 also decreases. Therefore it becomes easier to manage the metadata and it helps in increasing performance of a Hadoop cluster. Another reason for larger block size in Hadoop is to minimize the wastes of seek time. For a large data block the time required to transfer the data from disk is significantly longer than the seek time to locate the start of the block. This benefits over seek time is not possible in case of large number of blocks with small size.

Map Reduce Data Processing

MapReduce is a parallel data processing technique and it stressed on computation based on data locality. MapReduce fits well for processing mass volume of data stored across large number of nodes in a cluster similar to HDFS. The main idea of MapReduce is to bring computations to the data nodes where the involved data are present, instead of bringing data to some nodes which are ready for computation. Preventing unnecessary data movement by computing data at local nodes saves a lot 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 finally processed by reduce tasks. MapReduce frame work usually 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 suitable reduce tasks. Fig. 3 shows the data flow in a simple map reduce computation.

Drawbacks and Enhancements

Though Hadoop is a popular and famous Big Data analysis frame work, it is not free 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. Actual 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. Therefore master node is a single point of failure in earlier release of Hadoop. Though Master node is a high configuration machine, still its probability of failure cannot be eliminated. To overcome single point of failure, a secondary name node is introduced. It only keeps duplicate copy 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 copied. Thus the recovery process takes some time. So long recovery time is a major problem in this approach.

In later release 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 takes over its duties in a very small interval of time. There involves no delay of coping metadata as both the name nodes keeps updated metadata. However to maintain two name nodes there incur some additional overheads like updating metadata to both the name node besides the cost incurred in purchasing two high configuration machines.

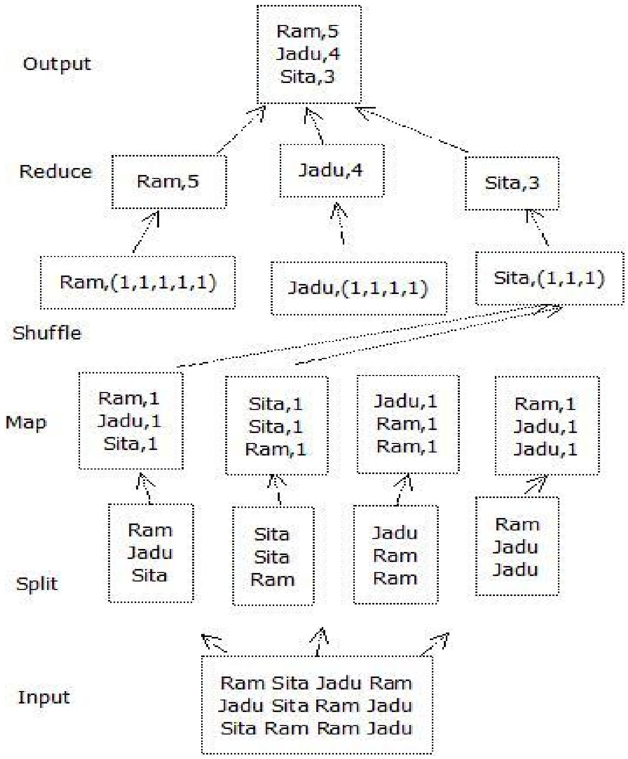

Fig. 3 Hadoop-MapReduce

In Hadoop when a user submits a Job it is completed 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 actual computations. When a user submits a job, a Job Tracker which runs on name node distributes the Jobs to many Task Trackers as small 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 increasing the performance reduces drastically after a certain limit of approximately 4000 nodes. To overcome this limitation there should some changes 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 additional 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 support more computation like graph processing besides MapReduce.

Big Data Impact on Society

Emergence of Big Data technologies made it possible for a wide range of people including researchers from social science and humanities, educational institute, government organization, and individual to produce, share, organize and interact with large scale data. With what motive and perspective do people from different groups use mass volume of data using latest technology is crucial. If it is used for decision making or opinion making or enforcement of new policies, it will have considerable long term impact on society and individual. The market sees Big Data as pure opportunity to target advertising towards right kind of people, which may bother an individual with flood of advertisements. Business and governments may exploit Big Data without concern for issue of legality, data quality. This may leads to poor decision makings. The threat of use of Big Data without a legal structure and strict law can hamper both individual and society as a whole.

Big data does not always mean as better data. A few Social scientists and policy maker sees big data as a representative of society. Which is not necessarily be true as a large portion of world population still does not dump data into Big Data repository by using internet or by any other means. 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 global population. More over number of accounts on social networking sites does not necessarily represent same number of people, as individuals can fake their identity and can create multiple accounts. A large mass of raw information in form of Big Data is not self-explanatory. And the specific 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 biased by some ethnic views or personal opinions.

Personal data can be sensitive and may have some privacy issue. It is valid and serious issue whether privacy can be maintained with increasing storage and usages of Big Data. For example there are huge data on health care system available today which can are used extensively for research purpose. And an individual can be identified from it and can be monitored periodically who is suffering from a disease without his or her knowledge. But it may emotionally or socially harm the person once his or her health information made public by people with evil intention. Many dataset contains identifier for individual such as name, date of birth or unique code issued by government agencies. So an individual can be spied with good or bad intention. Big data aggregator assumes 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 take over, the data set may go to some other hand and any existing privacy protection policy are unlikely to survive in a hand of a new owner.

CONCLUSION

A technology is not good or bad in itself. To use it for the welfare of society, Big Data operator must be denied a free ride by enforcing strict law and privacy policy to prevent misuse of data for wrong intention. The Big Data technology is new research area and being developing from last decade and there are scopes for improvements. It is playing an important role for global economy, scientific research, enforcement of social welfare scheme, and crime detection. Importance of Hadoop, NoSQL data base is increasing as RDBMS and SQL are not suitable to handle unstructured real 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 Internet Inst., Sept. 2011; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1926431.