Survey Paper on Big Data Processing and Hadoop Related Technologies

Mahammad Shabana
Research Scholar,
Dept of CSE,
CSJ MU, Kanpur.

Dr. Ravindranath
Professor,
Dept of CSE,
CSJ MU, Kanpur.

Abstract:

Today, an extensive amount of data is being continuously produce in all walks of life by all variety of devices and systems every day. Big Data deal with huge volume, complex, bigger data sets with number of, self-governing sources. With the rapid development of networking, data storage and the Data capturing capacity, Big Data are now quickly enlarging in engineering domains, including physical and biological sciences. A notable portion of huge data is being captured, stored, aggregated and analyzed in a efficient way. This the widespread background of big data and data analysis related technologies, such as Hadoop and MapReduce. We focus on the five stages of the value chain of big data, i.e., data creation, data acquisition, data storage, and data analysis. For each stage, we initiate the general background, express the Hadoop related technologies like Hive, Pig, Ambari, ZooKeeper, HBase. These Technologies handled huge amount of data in Terrabyte, Petta byte, Exa byte.

Keywords:

Big Data, Hadoop, HDFS, MapReduce, Pig, Hive, HBase, ZooKeeper, Ambari.

Introduction:

Right now we are living in Data world. Everywhere we are seeing only data. So, important thing is that how to store the data & how to process the data. 'Big Data', the term itself suggests huge amount of data. So the data which is beyond to the storage capacity and the data which is beyond to the processing power are called Big Data. Different data generator factors are their which can generate huge data. For example, New York Stock Exchange generates about one terra byte of new trade data per day.

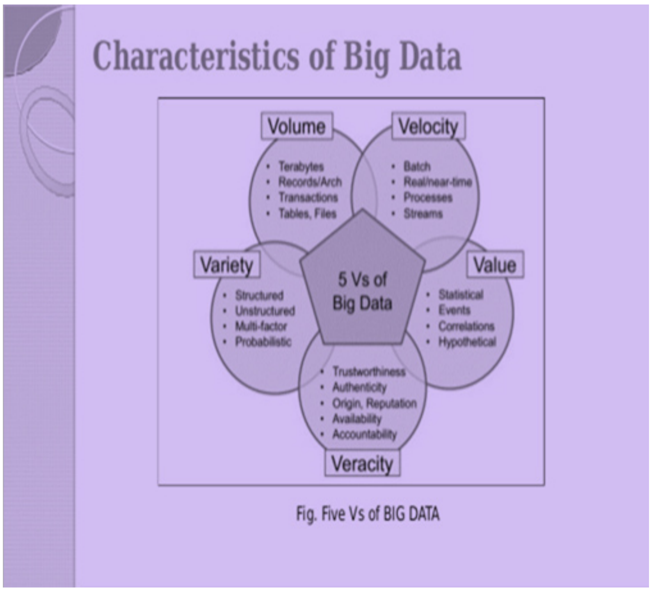
The large hydra collider near Geneva, Switzerland will produce about 15 peta bytes of data per year. Data centre's are required to store enormous amount of data .The administrator of data centre will maintain servers either IBM servers or EMC servers, called Sand Boxes. Hadoop is the best solution for processing Big Data. Hadoop consists of two vital components HDFS and MapReduce Framework.

HDFS is recommended to store large data sets and MapReduce is to process those large data sets which is stored on the HDFS. HBase is a Non-relational, distributed database system which is written in Java. It runs on the top of HDFS. It can serve as the input and output for the MapReduce. A platform for big data processing is pig. Pig appends one more level abstraction in data processing and maintaining data processing jobs so easily.

Hive is a dataware housing framework placed on top of Hadoop. HiveQL allows to write SQL like queries to process and analyze the big data which is stored in HDFS. Sqoop is tool which can be used to transfer the data from relational database environments like oracle and mysql. Sqoop is a commandline interface platform that is used for shifting data between relational databases and Hadoop. Zookeeper is a centralized service which produces distributed synchronization, group services and it also maintains the configuration information etc.

Characteristics of Big Data:

IBM has given three characteristics of Big Data. Volume (size), Velocity (Speed) and Variety (all types of data such as structured and unstructured data). These three terms together called Big Data, simply called 3V's. Now a days the 3V's are enhanced to 5V's.



Characteristics of Big Data

Fig. Five Vs of BIG DATA

Volume:

Volume refers to the amount of data. Volume represents the size of the data. How the data is enormous. Size of the data is rapidly increasing terabytes, pettabytes or exa bytes.

Velocity:

Velocity refers to the speed of the data processing. Data fetching and sending problems from local machine to data centre and data centre to local machine is made complex. Data comes at high speed. Big data is time sensitive.

Variety:

Variety makes data too big in size. The files come in variety formats and of any type. Data may be structured, unstructured and semi-structured data. Relational Database Management system manages structured data only. Social Networks like Face Book deliver unstructured data like video, images, text messages, audio etc. Log files come under semi-structured data.

Value:

Value is an important source of big data. The potential value of big data is extra large. It takes an important role for industries, organizations to store big data of values in database.

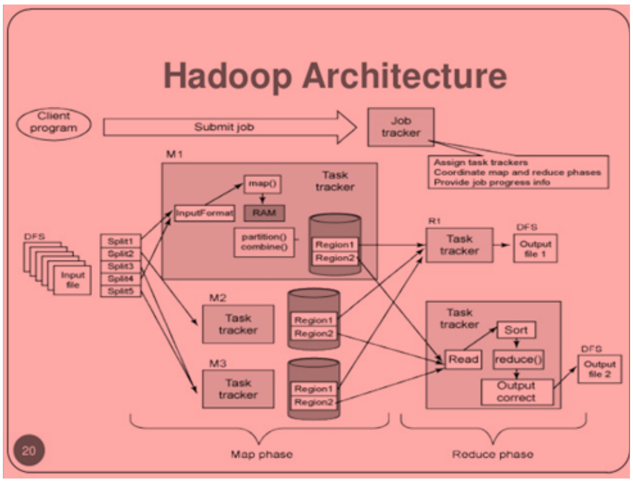
Veracity:

Veracity represents to biases and noise. All the data are not giving 100% correct results, when we dealing with 3V's. It consist dirty data.

Hadoop:

To enhance the processing speed for large datasets, Hadoop concept is introduced. Hadoop very well known how to store and process huge data sets with less processing speed. Hadoop is an open source framework written in Java, given by Apache Software foundation for storing and processing huge data with cluster of commodity hardware. Hadoop is basically used for storing and processing huge data sets but not recommended for small data sets. It is constructed to scale up from single servers to thousands of machines, each contributes local computation and storage. Two core concepts of Hadoop are Hadoop Distributed File System (HDFS) and Map Reduce.

Hadoop Architecture:



Hadoop Architecture

HDFS:

HDFS is a specially designed File System for storing enormous data sets running clusters on commodity hardware with streaming data access pattern. In normal File system each block size is 4KB. One of the disadvantages in normal file system is wastage of memory, which can be avoided in HDFS. To design HDFS, install the Hadoop with HDFS on top of normal file system. By default, each block size in HDFS is 64 MB. That's why it is called specially designed file system. If we uninstall HDFS, it will be converted as normal file system. Hadoop administrator will take the responsibility to maintain the HDFS. HDFS architecture consists of Master Daemons and Slave Daemons. Name Node, Secondary Name Node and Job Tracker are called Master Daemons. Data Node and Task Tracker are called Slave Daemons.

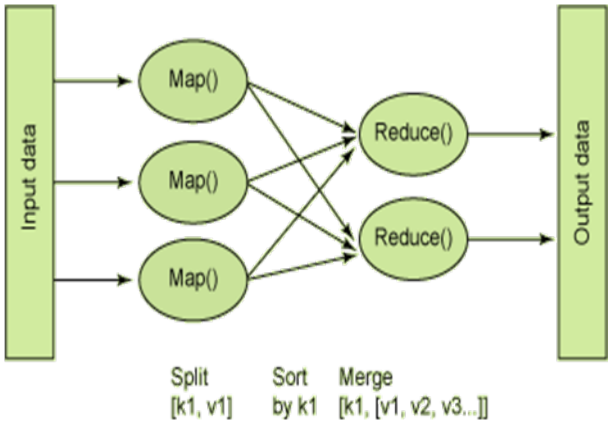
Name Node works as Master Node. Name Node is the heart of an HDFS file system. It keeps the directory tree of all files, and tracks where across the cluster the file where data is kept. It does not store the data of these files itself. Client applications talk to the Name Node whenever they want to place a file, or when they want to add/copy/move/delete a file. The Name Node answers the successful requests by returning a list of applicable Data Node servers where the data lives.The Name Node is a Single Point of Failure for the HDFS Cluster. HDFS is not a highly reliable system. When the Name Node goes down, the file system goes offline. There is a Secondary Name Node which is optional that can be hosted on a separate machine. It only creates checkpoints of the namespace by combining the edits file into the fsimage file and does not provide any real redundancy. Hadoop 0.21+ has a Backup Name Node that is part of a plan to have an High Availability(HA) name service, but it needs active contributions from the client who want it (i.e. you) to make it Highly Available.

It is essential to look after the Name Node. Here are some recommendations from production use:

- Use a good and reliable server with lots of RAM. The more RAM you have, the larger the file system, or the smaller the block size.
- Use Error Correcting Code RAM.
- On Java6u15 or later, run the server Virtual Machine with compressed pointers -XX:+ Use Compressed Oops to cut the Java Virtual Machine heap size down.
- List more than one name node directory in the configuration, so multiple copies of the file system metadata will be stored. As long as the directories are on separate disks, a single disk failure will not corrupt the metadata.
- Configure the Name Node to store one set of transaction logs on a separate disk from the image.
- Configure Name Node to store another set of transaction logs to a network mounted disk.
- Check the disk space available to the Name Node. If free space is attaining low, add more storage space.
- Do not host Data Node. Job Tracker or Task Tracker services are on the same system.

MapReduce:

Heart of the Hadoop is MapReduce Framework. It is given by Apache Hadoop software foundation. It is a programming model which enables resilient, distributed processing of gigantic unstructured data sets across clusters of commodity hardware. Each node of the cluster includes its own storage space. The term MapReduce actually refers to two separate and well defined tasks that Hadoop programs perform. During the map phase, the input file/data is divided into input splits for analysis. Map tasks running in parallel across the Hadoop cluster. By default, the MapReduce framework acquires input data from the Hadoop Distributed File System (HDFS). Using the Mark Logic Connector for Hadoop enables the framework to procure input data from a Mark Logic Server instance. The reduce phase uses whatever results getting from map tasks as input. The reduce tasks consolidate the in data into final results. The MapReduce framework stores results in HDFS by default. By using the Mark Logic Connector for Hadoop enables the framework to store results in a Mark Logic Server instance. However, the reduce phase depends on output from the map phase, map and reduce processing is not necessarily sequential manner. That is, as soon as any map task completes reduce tasks can begin. Before any reduce task can begin it is not necessary for all map tasks to complete. MapReduce works on key-value pairs. Conceptually, a MapReduce job gets a set of input key-value pairs and delivers a set of output key-value pairs by passing the data between map and reduce functions. The map tasks generate an intermediate set of key-value pairs that the reduce tasks uses as input. The below diagram illustrates the succession from input key-value pairs to output key-value pairs at a high level.



Although each set of key-value pairs is homogeneous, the key-value pairs in each step need not have the same data. For example, the key-value pairs in the input set is (KV1) can be (string, string) pairs, with the map phase induces (string, integer) pairs as intermediate results

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

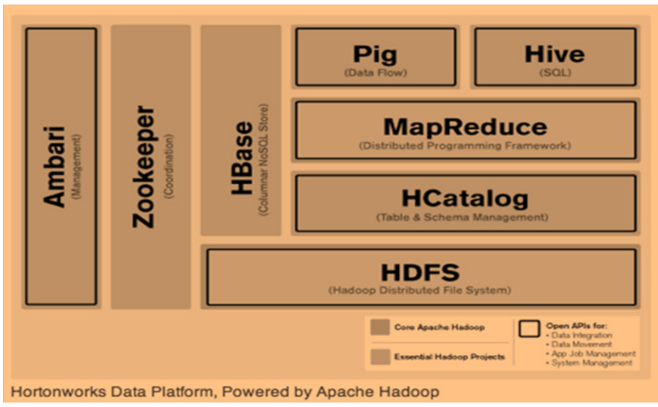
(KV2), the reduce phase producing (integer, string) pairs and for the final output results (KV3). The keys in the map output pairs need not be distinctive. In between the map processing and reduce processing, a shuffle sorts all map output values with the same key into a single minimize input (key, value-list) pair. The 'value' is a list of all values sharing the same key. Thus, the input to reduce task is actually a set of (key, value-list) pairs.The types of key and value at each stage determine the interfaces to map and reduce functions. Therefore, before coding a job, decides the data types needed at each stage in the map process and reduce process. For example:

1.Choose the reduce output key and value types that represents the desired outcome.
2.Choose the map input key and value types best suited to represent the input data from which to acquire the final result.
3.Determine the transformation necessary to get from the map input to the reduce output, and select the intermediate map output or reduce input key value type to match.

To Control MapReduce job characteristics through configuration properties, job configuration specifies:

- how to collect input
- the types of the input and output key-value pairs in each stage
- the map and reduce functions
- how and where to store the final results

Hadoop Ecosystem:



HCatalog:

HCatalog permits users with different data processing tools (such as Apache Hive, Apache Pig, MapReduce) to share data on the Hadoop cluster in an easier manner. It covers HCatalog's chief motivation, goals, the most essential features, currently supported file formats.

Pig:

Pig consists of a language. Pig's language, called as PigLatin, is a data flow language - this is a kind of language in which we program by connecting things together. Pig can work on complex data structures, yet those that can have levels of nesting. Unlike SQL, Pig does not necessitate that the data must have a schema, so it is effectively suited to process the unstructured data. But, Pig can still hold the value of a schema if we want to supply one. Relationally PigLatin is complete like SQL, which means it is at least as strong as a relational algebra. Turing completeness involves conditional constructs, an infinite memory model, and looping constructs. PigLatin can be Turing complete when it is extended with User-Defined Functions.

Hive:

Hive is a powerful technology developed at Facebook that revolves Hadoop into a data warehouse complete with a dialect of SQL for querying. Being a SQL dialect, Hive Query Language(HiveQL) is a declarative language. In PigLatin you specify the data flow, but in Hive we expore the result we want and Hive figures out how to erect a data flow to achieve that result. Unlike Pig, in Hive a schema is required, but it is not limited to only one schema. Like PigLatin and the SQL, HiveQL itself is a relationally complete language yet it is not a Turing complete language. It can also be extended through UDFs just like PigLatin to be a Turing complete. Hive works in terms of tables. There are two kinds of tables you can create: managed tables whose data is managed by Hive and external tables whose data is managed outside of Hive. Another option Hive provides for speeding up queries is bucketing. Like partitioning, bucketing splits up the data by a particular column, but in bucketing you do not specify the individual values for that column that correspond to buckets, you simply say how many buckets to split the table into and let Hive figure out how to do it.

HBase:

HBase is distributed database is an open-source, NoSQL, distributed, column-oriented store that runs on top of HDFS and is developed as part of Apache's

Hadoop. HBase is really more a "Data Store" than "Data Base". An HBase system consists of a set of tables. It can be composed of Table name, row key, column family, columns and time stamp. The row is uniquely recognized with the help of row keys and time stamp and the column family are static and columns are dynamic. If the application having variable schema and each row is slightly different then HBase is preferable. Thus the HBase cloud architecture could able to handle random real-time reading and writing of Big data thus the there is increase in performance than using Hadoop. Utilization of cluster resources is high because of Memory store usage. Thus image processing cloud is able to handle most familiar languages and it also helps to increase the computation speed.

ZooKeeper:

Zoo keeper instance is started along with the HBase. It keep track of all region servers in HBase. It gives information about how many region servers are there and which region servers are holding from which data node to which data node. One of the specialty is it keeps track of smaller data sets which Hadoop does not do. The HMaster contacting Zoo keeper it gets the details of region servers.

Ambari:

Apache ambari is a completely open source operational framework for provisioning, managing and observing apache hadoop cluster. Ambari contains an intuitive collection of operator tools and a set of APIs that hidden the tough task of hadoop. It compact the operation of cluster. With hundreds of years combined experience of Hortonwork along with the hadoop community have responded the call to produce the key services required for enterprise hadoop. It does ambari enables the system administrators to organize, monitor and to provision a hadoop cluster and it is also used to integrate hadoop with infrastructure of the existing enterprise.

Sqoop:

Apache Sqoop is a tool which is well organized for efficient export or import of bulk data between Apache Hadoop and structured datastores. Sqoop is a Top-Level Apache project which is a command line interface application written in java. The purpose of Sqoop is designed efficiently for the purpose of transferring huge amount of data between hadoop and structured data stores such as relational. It copies data quickly from external systems to hadoop. It enables data transfers from external data stores and enterprise data warehouses into hadoop. It ensures fast performance by parallelizing data transfer and uses optimal system. Sqoop supports analyses of data efficiently. It even mitigates excessive loads to external systems.Sqoop is used to exchange the data between Hadoop and IBM Netezza data warehouse appliance. The Apollo group, Education Company uses Sqoop to retrieve data from databases and to inject the results into relational database from hadoop jobs. In addition to this, there are countless other hadoop users who use Sqoop to efficiently transfer their data.

Avro:

Avro is a data serialization format which leads data interoperability among mutlple components of apache hadoop. Most of the components in hadoop supporting Avro data format. It works with basic premise of data produced by component should be readily consumed by other component Avro consists of so many main features Rich data types, Fast and compact serialization. It can support many programming languages like java, Python.

Mahout:

A library for machine-learning and data mining is Mahout. It is classified into four main groups: collective filtering, categorization, clustering, and mining. The Mahout library belongs to the subset that can be executed in a distributed mode and also it can be executed by MapReduce.

Oozie:

The Oozie component produce features to manage the workflow and dependencies, eliminating the need for developers to code custom solutions.

Conclusion:

Nowadays, orgaizations need to process Multi Pettabyte Datasets efficiently. The Data may not have strict schema for the large system. It has become more Expensive to build reliability in each Application for processing petabytes of datasets. If there is problems of Nodes fail every day, some of the causes of failure may be. Rather than exceptional, failure is expected.

The number of nodes in a cluster is increased more in number. So there is a Need for common infrastructure to have Efficient, reliable, and Open Source Apache License. In this manner we discussed about the basics of Big Data and Hadoop distributed file system.. The paper also focuses on Big Data storage and processing problems. These technical challenges must be directed for efficient and speed processing of Big Data. In this paper we have tried to cover all detail concepts of Hadoop and Hadoop component and future scope.

References:

[1] Apache Hadoop: http://Hadoop.apache.org

[2]http://developer.yahoo.com/hadoop/tutorial/module1.html

[3]http://iasir.net/AIJRSTEMpapers/AIJRSTEM13-131.pdf

[4]An Oracle White Paper, "Hadoop and NoSQL Technologies and the Oracle Database", February 2011.

[5] http://hortonworks.com/hadoop/ambari/

[6]http://link.springer.com/content/pdf/10.1007%252F978-1-4302-4864-4_20.pdf

[7] http://www.tutorialspoint.com/sqoop/sqoop_pdf

[8] A. Pavlo et. al. A Comparison of Approaches to Large-Scale Data Analysis. In Proc. of ACM SIGMOD, 2009.

[9]Hive Performance Benchmark. Available at http://issues.apache.org/jira/browse/HIVE-396

[10] TPC-H Benchmark. Available at http://www.tpc.org/tpch

[11] Jeffrey Dean and Sanjay Google, Inc." MapReduce: Simplified Data Processing on Large Clusters"

[12]Vasiliki Kalavri, Vladimir VlassovKTH The Royal Institute of Technology Stockholm, Sweden kalavri@kth.se "MapReduce: Limitations, Optimizations and Open Issues". TrustCom/ISPA/IUCC,Page1031-1038,IEEE,(2013)

[13] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu and Raghotham Murthy "Hive – A Petabyte Scale Data Warehouse Using Hadoop" By Facebook Data Infrastructure Team

[14] Apache HBase. Available at http://hbase.apache.org