Recent Trends in Big Data Technologies

Pallavi Sood, Vaishali Gupta

PG Department of Computer Science & IT, Lyallpur Khalsa College, Jalandhar,Punjab,India

Abstract- The size of data is widening very rapidly because of increasing usage of internet, smart gadgets and social networking sites and maximum data is multi-structured. The traditional database models or technologies are unable to meet the needs of organization's heavy workload. So to process large volume of multi-structured data, several companies like facebook, twitter, google deals with big data problems so they developed their own non-relational systems. Finance and Insurance sectors, telecommunication companies and other data centric companies have large datasets for a long time. Big data analytics provide new means for businesses and government to analyze huge volume of structured, semi-structured and unstructured data. New types of data will give new challenges and opportunities as well. Organizations are becoming more flexible and more open day by day. The challenges include capture, storage, cleaning, curation, integration, search, processing, transfer, indexing, mining and visualization. These days big data is one of the most debatable topic in IT industry. It is going to play a key role in near future. Big data changes the way the data is managed, used and permit organizations to handle certain types of workload that was not possible earlier. This paper shows technologies used for big data.

Keywords- *Big data; Technologies; Hadoop; Map Reduce*

I. INTRODUCTION

We can't imagine a world without data storage, a place where data is stored that may be retrieved at a later time. Today, web has been overloaded with huge amount of data.The amount of data on web is measured in exabytes(10 pow 18) and zettabytes (10 pow 21). The amount of available data has been increased from the past few years because of new social attitude, societal transactions as well as the rapid speed of software systems.Big data has become an essential driver for innovation and growth that relies on innovative technologies such as cloud computing, internet of things and analytics. The usage of data is rapidly changing the nature of communication, shopping, entertainment, advertising and relationship management. Scientists regularly face problems because of huge data sets in various fields including internet search, biological environmental research, physics simulations, organic and inorganic chemistry and business & finance informatics.Big data usually comprises of data sets with sizes which are beyond the reach of commonly used software tools to acquire, create, manage and process the data within a bearable elapsed time. A major investment in big data that is properly organised, can result not only in major scientific advancements, but also lay the foundation for the next generation of advancements in science, technology, medicine and business.The paper is divided in the following sequence: Starting with the introduction, we discuss about the features of big data (5 V's) and then it is followed with technologies used in big data.

Big Data: Big data is a term used for data sets that are very large in size and handles structured, semi-structured and unstructured data. Such kind of data is difficult to store by traditional database systems. Big data is expected to grow $34.7 by 2018. Five main features characterize big data: Volume, Variety, velocity, Value and Veracity.

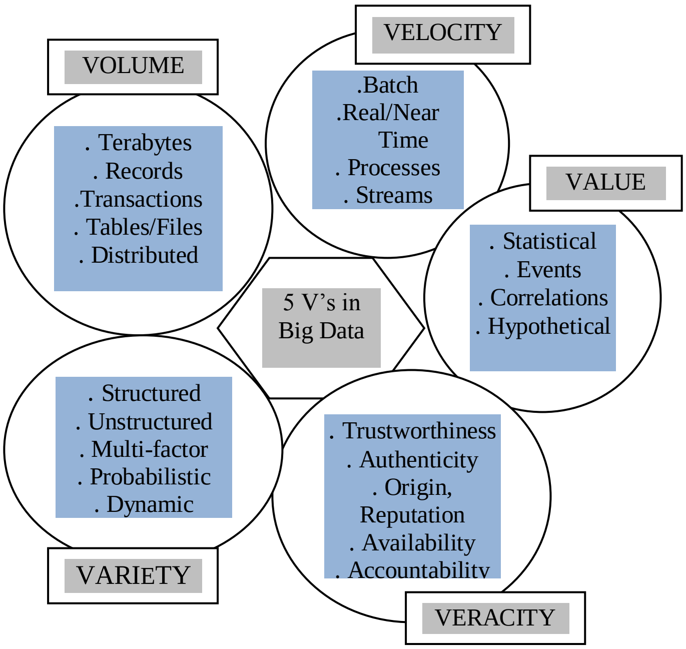


Fig.1: 5 V's of Big Data

A. Volume (Data in Rest)

Organizations gather data from a variety of sources including business transactions, social media, information from sensor, health care data, financial data and machine-to-machine data etc. In the past, storing it would've been a problem but now technologies (such as Handoop) have erased the burden.

B. Variety (Data in Many Forms)

The data does not posses a fixed structure. Data may exist in a variety of file types, including highly structured data(data from Relational databases), semi-structured(Web logs, Social media feeds, email etc) or unstructured(still pictures, video, audio, clicks & streaming data from sensors).

C. Velocity(Data in Motion)

Big data velocity cope with the rate at which data flows in from sources like business processes, machines, networks and human interaction with things like social media sites, mobile devices etc. The flow of data is heavy and continuous. This real time data can help researchers and businesses make important decisions that provide critical competitive advantages.

D. VALUE (DATA IN HIGHLIGHTS)

Value is the important aspect in big data. Though big data has large volumes and wide variety of data, but data has no use if it does not have any value.

E. Veracity (Data in doubt)

When we are dealing with high volume, velocity and variety of data, it is not possible that all of the data is going to be 100% correct, there will be unusable data also. The quality of data being acquired can vary greatly. The data accuracy of analysis depends on the veracity of the source data.

II. TECHNOLOGIES USED IN BIG DATA

A latest survey says that 80% of the data created in the world are unstructured. One challenge is how this unstructured data can be structured, before we try to understand and occupy the most important data. Another challenge is how we can save it. To fulfill such challenges various technologies are developed. These technologies can be classified into two categories: Storage & Querying/Analysis.

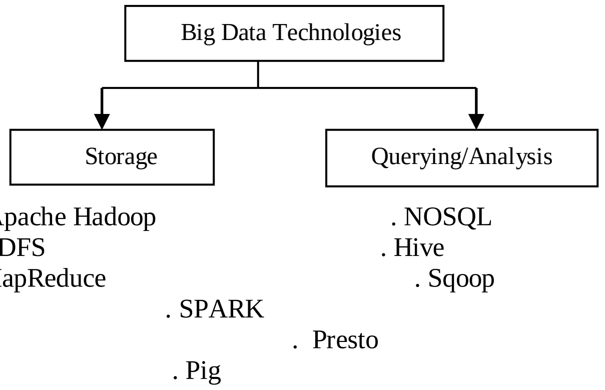


Fig. 2: Technologies of Big Data

III. STORAGE TECHNOLOGIES

A. Apache Hadoop

Hadoop is an open source, Java based programming structure that guides the handling and storage of extremely huge data sets in a distributed computing domain. It is part of the Apache project sponsored by the Apache software foundation.

Hadoop makes it achievable to run applications on systems with thousands of commodity hardware nodes, and to handle thousands of terabytes of data. If we talk about traditional approach, Big data was processed by powerful computers. But these computers had processing limits and these computers were not scalable. But in Hadoop's approach, data is broken into pieces and send to various computers for computations. All these computations takes equal quantum of time. After computations, all the results are merged together and final result is send to big data.

All the computers that works as a Slave computer on Hadoop has two components:- Task Tracker, Data Node.

Task Tracker:- The Task tracker's job is to handle the smaller pieces of tasks that has been given to a particular node.

Data Node:- The job of data node is to manage the piece of data that has been given to a particular node.

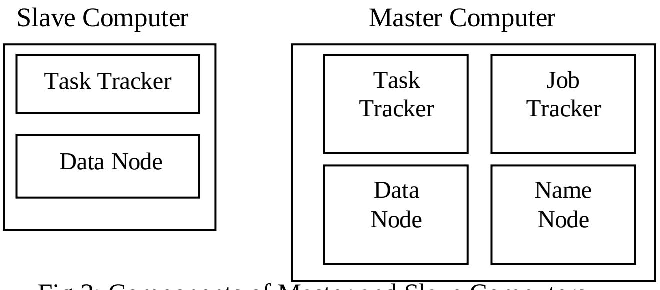


Fig.3: Components of Master and Slave Computers

All the computers that works as a Master computer on Hadoop has four components:- Task Tracker, Data Node, Job Tracker and Name node.

Applications running on Hadoop will contact to the master node. It is a batch processing tool so that applications would assign or provide tasks to hadoop in the form of queue. Once the task is completed, applications will be informed and results would given back to the applications.

Job Tracker:- The job tracker component running on master node is to split the higher bigger tasks into smaller pieces and to send each small piece of data/computation to Task tracker. The task tracker works on these smaller pieces and send result back to the Job tracker and Job tracker and then it sends the combined results to the application.

Name Node:- The name node running on master computer is responsible to keep an index of which data is residing on which data node. When application contacts the name node, it tells the application to go to this particular computer and get data.

In Hadoop, programmers don't have to concern about where data is located, how to manage failures, how to break computations into pieces and how to program for scaling. Hadoop is highly scalable. It could consist of 1 computer and go upto 1000 computers. The number of computers depends upon the requirements and changing needs. The core of Apache Hadoop consists of a Storage part, known as Hadoop Distributed File system (HDFS), and a Processing part Which is a MapReduce programming model. Hadoop breaks files into large blocks and distributes them across nodes in a batch. It then transfers gathered code into nodes to process the data in parallel. This approach has an advantage of data locality, where nodes manipulate the data they have access to. This allows the datasets to be processed rapidly and more efficiently. Popular Hadoop vendors include Cloudera, Hortonworks, MapR and the leading public clouds all offer services that support the technology.

Areas where Hadoop is used :

[1]. Social Media
[2]. Retail Sector
[3]. Financial Services
[4]. Searching Tools
[5]. Government Sector
[6]. Intelligence Services etc
[7]. Companies like Google, Facebook, Amazon, EBay, American Airlines, The NewYork Times etc used Hadoop to manage their large data.

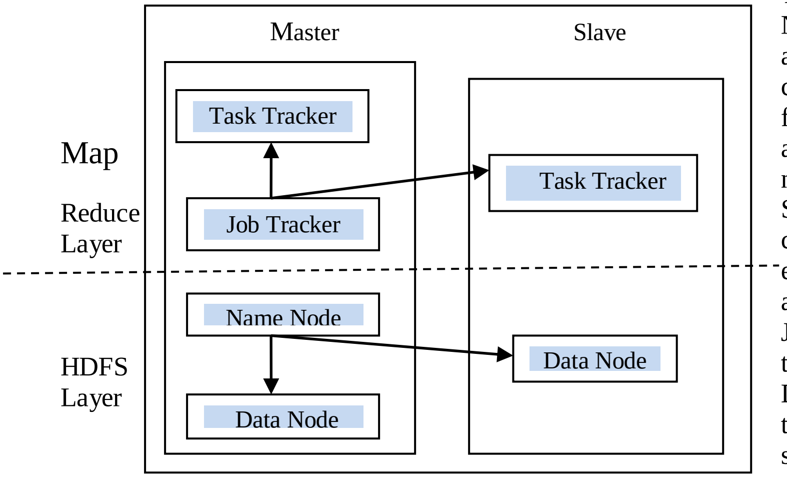


Fig.4: Architecture of Hadoop

A. HDFS

HDFS stands for Hadoop Distributed File System. It is specially designed for storing large data sets with batches of commodity hardware with streaming access pattern. Cluster is usually a single computer serves as a node, and when many such nodes are connected through LAN, then this system logically form a single unit. Commodity hardwares are the cheap hardwares. HDFS stores individual file as a sequence of blocks. Here all blocks in file except last block are of same size. HDFS is designed to work on large files, it would not work efficiently with lots of short files.

File system is that system which is used for storing files/directories in hard disk. Normally, Hard disk is split into number of blocks, by default size for block in 4KB. For example, Total size of disk is 500 GB. Suppose this 4KB block stores 2KB data and remaining 2KB block will not release for some alternative file and it will be wasted. But in HDFS, the remaining block is released for some alternative file and it is not wasted. Eg- if Block size= 64 MB, File size= 35 MB, Extra space=29MB and it will not be wasted in case of HDFS.

IV. ADVANTAGES

1) HDFS provides a reliable, expandable and manageable solutions for coping with huge amount of data.

2) It supports parallel reading and processing of data.

3) It supports read, write, rename and append operations. It is used for optimizing streaming read/writes of large files.

4) HDFS is built in fault tolerant and allows easy management.

5) It automatically manages addition/removal of nodes without any operational intervention.

6) It is used in production at companies such as Yahoo, facebook Twitter, Ebay etc.

HDFS uses Master Slave architecture. There are five service provided by HDFS are: Name Node, Secondary Name Node and Job Tracker are under Master Services and Data Node and Task Tracker are under Slave Services.

Name Node:- A HDFS cluster consists of a single Name node, a master server that handles the file system name space and controls access to files by clients.The name node implements file system operations like opening, closing and renaming file and directories. It also regulates the mapping of blocks to data nodes. Name Node maintains name space image and edit log.

Secondary Name Node:- It's main motive is to have checkpoints in HDFS, which helps name node to function effectively. It timely merges the Edit log & namespace so as to avoid name node to rush out of memory.

Job Tracker:- A job tracker is a node in the cluster that give tasks- map, shuffle and reduce operations to the task tracker.

Data Node :- In HDFS, files are partitioned into blocks and these blocks are typically of size 128 MB. The blocks are stored in the local storage as data nodes. In addition there are many Data nodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. The data nodes are liable for serving read & write requests from the file system's client. The data nodes also perform block creation, deletion & replication upon instruction from name node. The name node and data node are pieces of software devised to run on commodity machines.

Task Tracker:- A task tracker is a node in the cluster that accepts tasks- map, shuffle and reduce operations from the job tracker.

Every master service can communicate to each other and every slave service can communicate to each other. If name node is master node, then data node is slave node. If Job tracker is master node, then task tracker is slave node. Name node can communicate to data node and vice versa. But name node does not talk to task tracker. In the same way, job tracker can communicate to task tracker and vice versa.

a) MapReduce

Mapreduce is the heart of Apache Hadoop. mapReduce is a vast parallel processing technique for handling data which is scattered on a commodity cluster. The vital unit of information used in MapReduce is a key-value pair. All types of structured and unstructured data need to be translated to a basic unit before supplying the data to MapReduce model. In MapReduce, the reduce task is always performed after the map job. There are three stages in MapReduce: Map stage, Shuffle Stage and Reduce Stage.

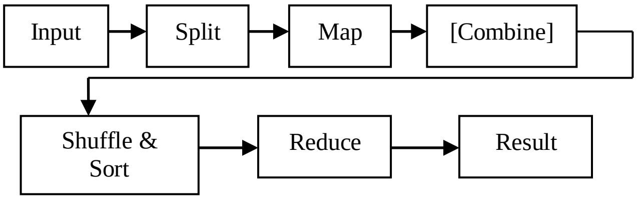


Fig.5: Logical Data Flow of MapReduce

Map Stage:- The mapper's task is the initial step of processing that processes each input record and generates intermediate key value pairs. Maps are the individual tasks that converts input records into intermediate records. The number of mappers usually relies on number of HDFS blocks (input splits) for the file. For each input split, a map task is created. So, over the lifetime of a map-reduce job, the number of map tasks matches the number of input splits. The generated key/value pair is entirely different from the original input pair.

Shuffle stage:- Shuffling is the process of swapping the intermediate outputs from the map tasks to where they are needed by the reducers.

Reduce Stage:- The reducer's job is to reduces a set of intermediate values which share a key to a smaller set of values. All the values that share the same key are presented to a single reducer together.

Real Life Example of MapReduce

Suppose we have score sheets of soccer games for a Team as Input.

Game=17, Date=230517, Goals=3, Ben=2, Tom=1
Game=18, Date=240517, Goals=1, Mike=1
Game=19, Date=250517, Goals=2, Ben=1, Mike=1
Game=20, Date=260517, Goals=1, Ben=1
Game=21, Date=270517, Goals=4, Tom=3, Mike=1
Game=22, Date=280517, Goals=1, Mike=1

Program Output - Total number of goals by individual players.

Limitations of MapReduce in Hadoop

1) Unsuitable in Real Time Processing- Being batch oriented, it takes minutes to executes jobs depending upon the volume of data and number of nodes in the cluster.

2) Unsuitable for Trivial Operations- For operations like Filter and Joins, you might need to rewrite the jobs, which becomes difficult due to the key-value pattern.

3) Unfit for huge data on network- However, it works on the data locality principal, it cannot handle a lot of data requiring shuffling over the network well.

4) Unsuitable with OLTP(Online Transaction Processing)- OLTP requires a huge number of small transactions, as it works on the batch oriented framework.

5) Unfit for Processing Graphs- The Apache Giraph library processes graphs, which adds further complexity on top of MapReduce.

6) Unfit for Iterative Execution- Being a state less execution, MapReduce does not suited with use cases like Kmeans that require iterative execution.

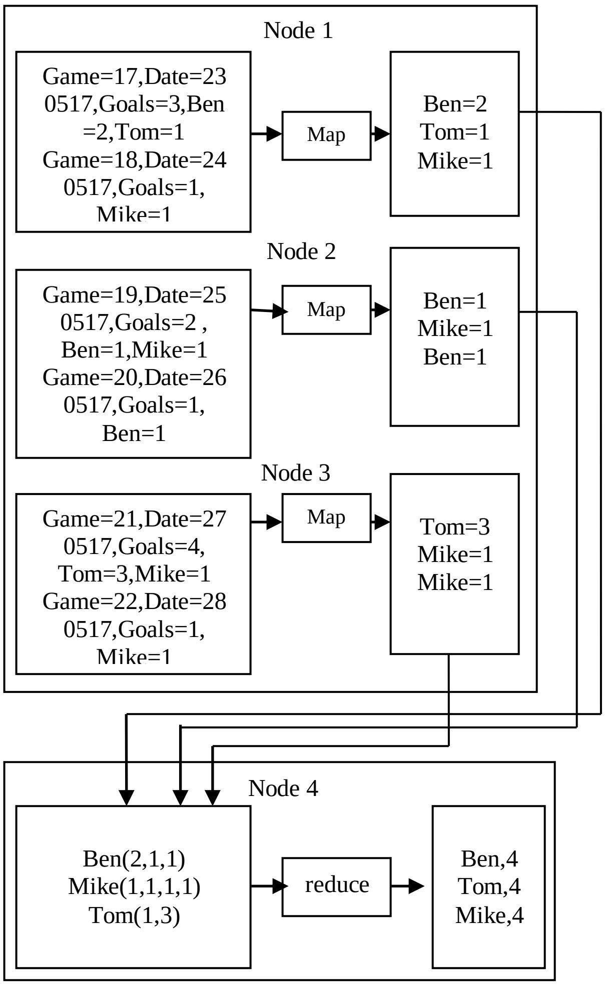

Node 1
Game=17,Date=23 0517,Goals=3,Ben =2,Tom=1 Game=18,Date=24 0517,Goals=1, Mike=1
Map
Ben=2 Tom=1 Mike=1
Node 2
Game=19,Date=25 0517,Goals=2 , Ben=1,Mike=1 Game=20,Date=26 0517,Goals=1, Ben=1
Map
Ben=1 Mike=1 Ben=1
Node 3
Game=21,Date=27 0517,Goals=4, Tom=3,Mike=1 Game=22,Date=28 0517,Goals=1, Mike=1
Map
Tom=3 Mike=1 Mike=1
Node 4
Ben(2,1,1) Mike(1,1,1,1) Tom(1,3)
reduce
Ben,4 Tom,4 Mike,4

Fig.6: Example of Program Flow

IV. QUERYING/ANALYSIS TECHNOLOGIES

These technologies are used to analyze huge amount of data and also provide facility to fetch data by querying on it. The technologies for Querying/Analysis include:

A. NOSQL

NOSQL stands for "NOT ONLY SQL". It is also known as Schema less databases. Schema is the data structure of a database relation, eg MYSQL. Every database relation should have a fixed data structure. Schema less implies database don't have any fixed data structure, Eg. NOSQL. Here you can change the data structure as you wish.

NOSQL is a non-relational database. It is a flexible database that is you can add any no. of records in it. The leading RDBMS vendors like Oracle and IBM now also offer NOSQL databases. NOSQL databases are becoming popular as the big data trend has grown. According to Allied Market Research, the NOSQL market could be worth $4.2 billion by 2020.

CAP Theorem: In CAP Theorem, C denotes Consistency, A denotes Availability and P denotes Partition Tolerance. We want that an ideal system should have all these properties. But according to CAP theorem, not all of C,A and P can be satisfied simultaneously. But RDBMS is not good for distributed systems means it does not satisfy P property of CAP, it only satisfy C and A property. It is the big criticism of RDBMS. That's by there is need of such database that can process distributed data. All NOSQL databases can satisfy P property of CAP theorem.

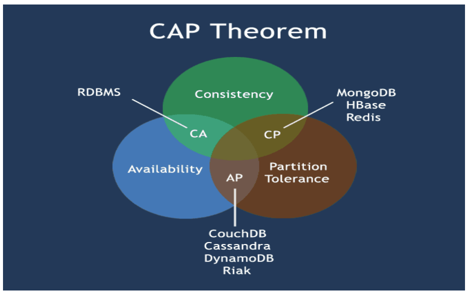


Fig.7: Components of CAP Theorem

NOSQL don't care about ACID properties, It care about BASE properties. BASE properties include:

Basically Available:- The system should guarantee of availability of data all the time.

Soft State:- The state is said to be soft state if the state may change without any input coming.

Eventually Consistent:- Data will be eventually consistent to all the nodes.

When to put into use NOSQL

[1]. The capability to store & retrieve great quantities of data is important.

[2]. The data is unstructured or the structure is modifying with passage of time.

[3]. Saving relationships between elements is not necessary.

[4]. Dealing with increasing list of elements like Twitter posts, Blogs, Internet server logs etc.

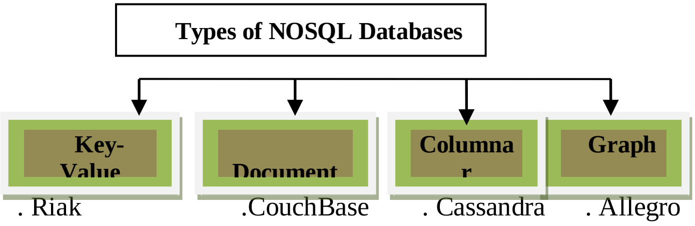
[5]. It is not required to be implement constraints and validations logic in database.

When Not to put into Use NOSQL

a) Complex Transactions need to be handled.

b) Joins must be managed by databases

c) Validations must be managed by databases.



. Riak	.CouchBase	. Cassandra	. Allegro
. Amazon Dynamo	. MongoDB	. HBase	.Neo4J
.Redis			.
OrientDB			
. Mem Cache			

Fig.8: Types of NOSQL Databases

B. SQOOP:

Sqoop stands for SQL + Hadoop= SQOOP.SQOOP=" SQL to Hadoop and Hadoop to SQL". It is created by Cloudera and then open sourced. It is single client program that interacts with hadoop file system to create one or more than one mapreduce programs. Sqoop uses primary key attribute to divide source data across its mappers. Sqoop works with anything that is JDBC complaint. It is a tool meant for efficiently transferring large amount of data between Apache hadoop and structured data stores such as relational databases. The RDBMS store structured data and passes structured data, if we are going to get large amount of structured data, sometimes RDBMS is unable to store and process that bulk of data, then we can store this data back in HDFS, because Hadoop give support of structured data. Keeping data from RDBMS to HDFS, needs some tool to support & this tool is sqoop. Sqoop is an interface in between RDBMS and HDFS so it works as a bridge between those two for either importing data or exporting data. Sqoop Import is the process of getting data in Hadoop from data sources. Sqoop Export is the process of taking data out of hadoop and put it into relational databases. There are range of connectors available to connect sqoop to traditional RDBMS such as Teradata, MYSQL, Oracle, Green Plum, Netteza, Micro Strategy etc. Sqoop is built on top of MapReduce. Generally Sqoop is running with sqoop interpreter. We know very well that Hadoop can run only on MapReduce. Even they are implementing sqoop comments. In client machine, sqoop interpreter will be running, When sqoop interpreter is going to be receive by the job tracker. The job tracker will internally commits the sqoop interpreter into mapreduce. By default, Sqoop is working with 4 mappers. If it is not functioning with reducers, then output will be directly given from the mappers. Working with 4 mappers means we will have 4 output files.

Features:-

1) Import entire table by using Sqoop.

2) Import part of the table with where clause or with columns clause.

3) Import all tables from a specific database.

4) Export a table from HDFS to RDBMS.

SQOOP Architecture

➔ Map only

➔ Command line only

➔ No client-server

If one can access to sqoop command, he will have access to all JDBC jars.

➔ Not easily extensible & no separation of duties.

Both read from source and write to target is done by mapper. Sqoop have many comments but most frequently used comments are Sqoop Import and Sqoop Export.

Sqoop Import:- Sqoop import is to get data from conventional databases and NOSQL/Document based databases into Hadoop eco- system. It uses MapReduce framework to store data in parallel.

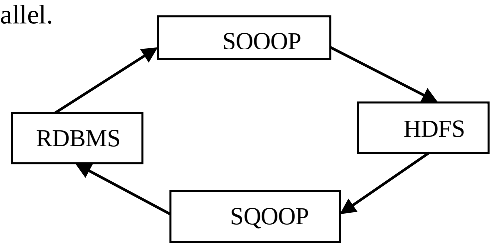


Fig.9: SQOOP Architecture

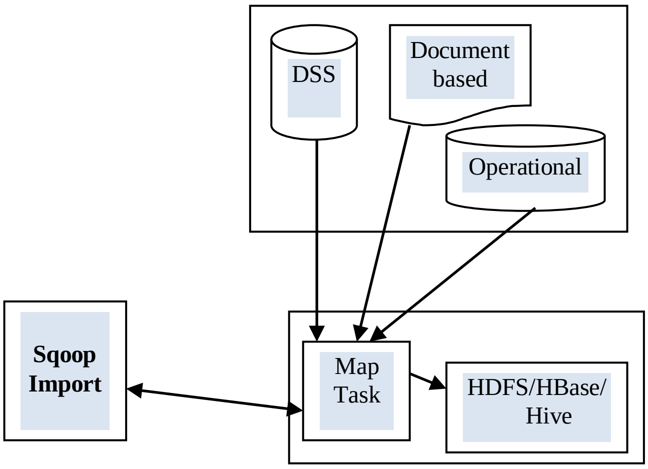


Figure 10: Architecture of Sqoop Import

Execution steps for Sqoop Import are

- Generates custom DBwritable class reading metadata of table.
- Metadata is the information about data like cardinality of relation, their names, data types etc.
- Connect to database. Default 4 concurrent connections are there.
- Read and split the data using custom DBwritable class.
- Store the data into HDFS.

Split logic:-

- Uses Primary key or unique key.
- Get minimum and maximum value.
- Compute ranges depends upon the number of map tasks (default4).
- Process mutually exclusive data in parallel.
- Without Primary/unique keys import process only uses one map task.

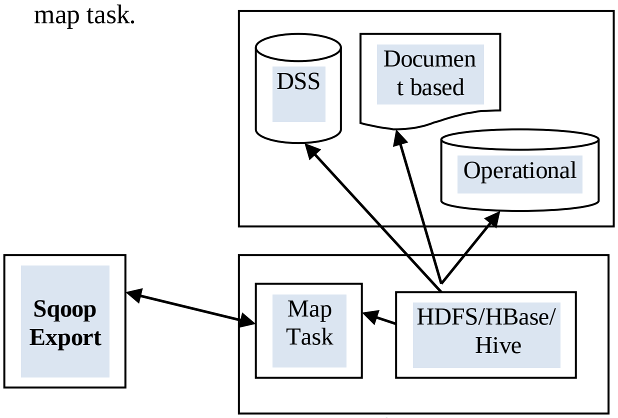


Fig.11: Architecture of Sqoop Export

SQOOP Export :- Sqoop export is to get data out of Hadoop based systems into conventional databases/NOSQL data stores. It also uses Map/Reduce framework. At this time it only understands HDFS directories not Hive tables(HCatalog). It also splits data(but uses HDFS splitable logic).

C. *PRESTO*

Presto is an open source distributed SQL query engine for executing interactive analytic queries against data sources of all sizes varying from gigabytes to petabytes. Query engine is a software component placed on the top of a data server that is responsible for implementing query functionalities against DB data and providing responses to users/applications. Query engine is liable for converting user queries into efficient data fetching and processing operations, as well as running these operations on single or multiple nodes in order to find answers to the queries. Interactive analytic is the capability to execute complicated queries across complex data landscapes where we have the complexity intelligence to traverse number of nodes in real time.

Presto is developed at facebook, later contribute to open source. Presto is build to deal with data warehousing and analytics: data analysis, aggregating bulk of data and generating reports. Presto has all the significant build in core functionalities- like parser, SQL functions and monitoring capabilities. Presto is written in highly tuned Java. Here data resides in memory during execution and is pipelined across nodes.

Key Differentiators

- Support for cross platform query capabilities.
- Support for Federated queries.
- Performance and scale.
- Used in productions at many well known web scale companies.
- Storage Independent.
- A big list of pluggable extensions.

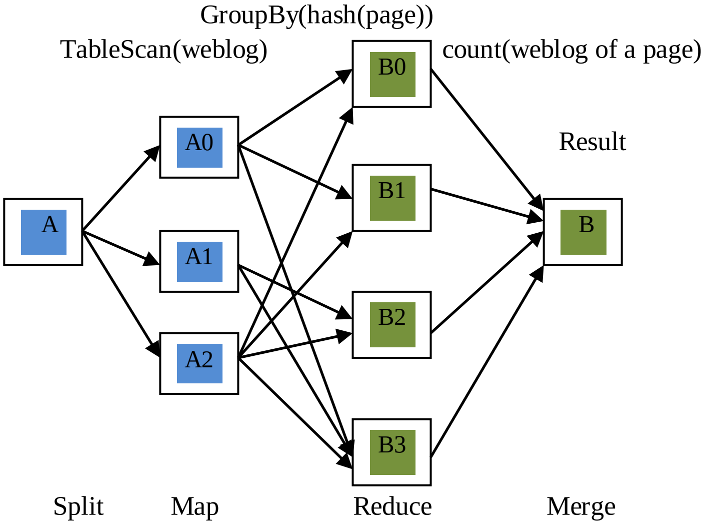


Fig.12: Presto Execution

Query Execution Model :- When presto scans a statement, it transforms it into a query and creates a distributed query plan which is then accomplished as a series of interconnected stages executing on Presto workers. A query implies the configuration and components instantiated to run SQL statement.

Stage:- When Presto runs a query, it does so by splitting the execution into a hierarchy of stages.

Tasks :- Stage is implemented as a series of tasks scattered over a network of presto workers. Tasks act upon or process splits.

Splits:- Tasks works on splits which are components of a huge data set. Stages at the lowest level of a distributed query plan to get data via splits from connectors.

Connectors:- Connectors are storage plugins that allow access to different data sources. Connectors are needed to impart interfaces for retrieving metadata, acquiring data locations and accessing the data. Presto gives a wide range of plugins to access different data sources.

Metadata API provides information about the relations and the attribute types.

DataLocationAPI gives information which is related to the physical location of blocks.

DataStreamAPI is responsible for acquiring data.

D. HIVE

Hive is a data warehouse infrastructure tool used to handle structured data in hadoop. It is devised on the top of Hadoop mapReduce framework so that it can be used to summate big data and make querying and analyzing data easy. Hive is developed by facebook , later by Apache software foundation took it up and developed it further so that they can make it open source under Apache Hive. Facebook analyzed several terabytes of data everyday using Hive. Hive is used by various companies like Amazon used it for Amazon Elastic MapReduce. There is no need to learn Java and Hadoop APIs for Hive.

Features

- Hive stores schema in a database and processed data into HDFS.
- It is developed for OLAP (Online Analytical Processing).
- It gives SQL type language for querying. We call it as HiveQL or HQL.
- Hive is fast, familiar and extensible.
- Hive is NOT
- A Relational database.
- It is not designed for OLTP (Online Transaction Processing).
- It is not a language designed for handling real time queries and row level updates.

In Hive,relations and databases are designed first and then data is stored in these tables for handling and querying structured data.

Hive has two components:

HCatalog:- It is basically a table or storage management layer for Hadoop. It allows users with various data processing tools. For example, MapReduce and Pig allows to read and write data more easily on the grid.

Web HCat:- It provides a service which can be used to run Hadoop mapReduce (or YARN), Pig , Hive jobs so that we can have metadata [illegible]tions using H[illegible].

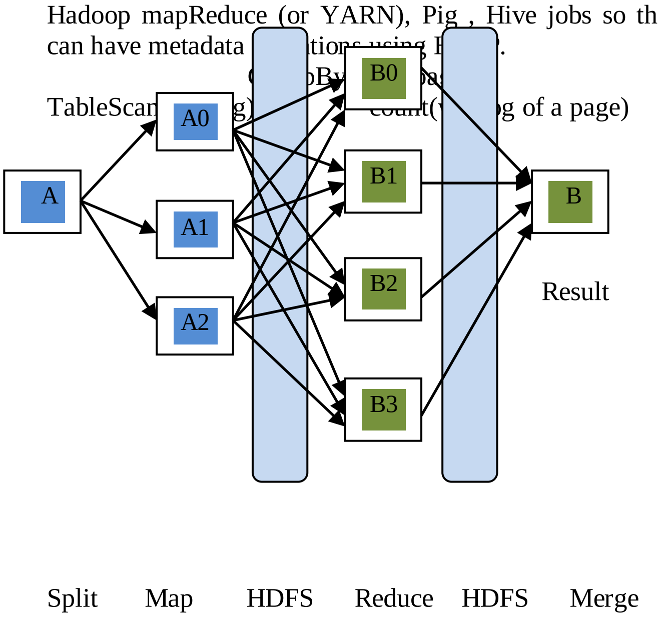


Fig.13: Hive Execution

Hive Architecture

User Interfaces:- Hive is a data warehouse infrastructure software. This software can create communication between user and HDFS. Hive comes with command line shell interface and used to design relations and run queries.

Meta Store:- Meta store is basically a database server where we can save schema or metadata table . It can support columns in table, datatypes and HDFS mapping.

HiveQL:- This is a language used for distributed queries on huge amount of data.

Hive Engine compiles HIVEQL/HQL queries into MapReduce jobs. To be executed on Hadoop. In addition, we have custom MapReduce scripts can also be plugged with queries.

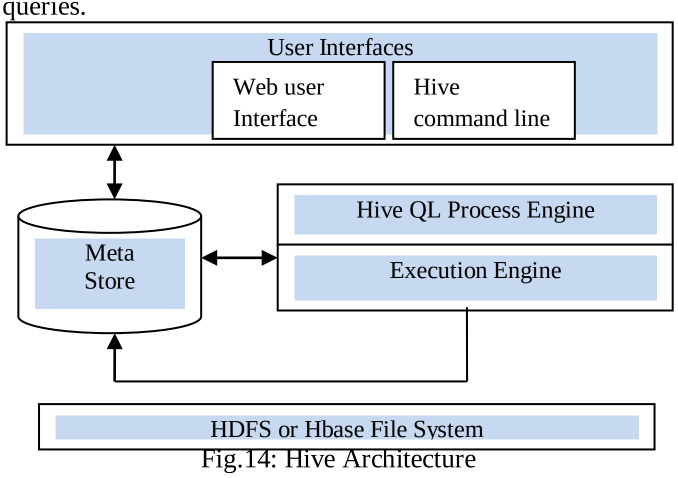


Fig.14: Hive Architecture

E. SPARK

SPARK is an open source cluster computing framework. It is suitable for real time processing, trivial operations and processing larger data on network. It is popular for its performance benefits over MapReduce. Spark provides upto

100 times faster performance for a few applications with in-memory primitives , in contrast to the two-stage disk based MapReduce paradigm of hadoop. It is suitable for machine learning algorithms, as it allows programs to load and query data repeatedly. Another important benefit is language flexi ibility. It supports various development languages like Java, Scala, Python and will likely support R. Every type of data processing can be performed using SPARK that you execute in Hadoop.

SPARK supports

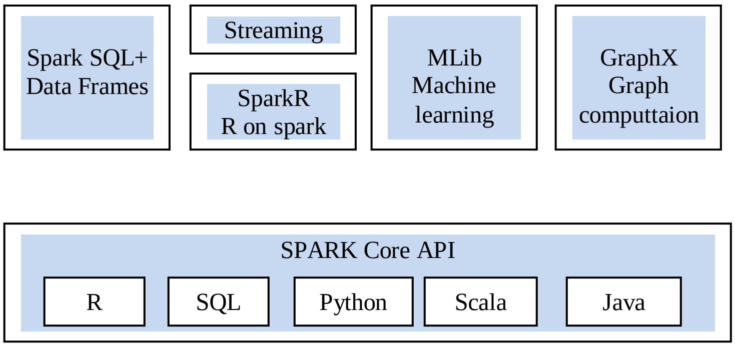
1) Batch Processing- Spark batch can be used over Hadoop MapReduce.

2) Structured Data Analysis- Spark SQL can be used using SQL.

3) Machine Learning Analysis- MLLib can be used for clustering, recommendation and classification.

4) Interactive SQL Analysis- Spark SQL can be used over impala. Impala is a huge parallel processing SQL query engine for handling extensive volume of data that is stored in Hadoop cluster.

5) Real Time streaming Data Analysis- Spark Streaming can be used over specialized library like storm. Apache Storm is a free and open source distributed computation system. It is designed for real time computations.



F. PIG

Pig is an open source high level dataflow system. It gives a simple language for querying and data manipulation. It is developed by yahoo, open sourced by Apache. It runs on Apache Hadoop YARN and also make use of HDFS and MapReduce and makes it viable to handle huge jobs to process large volumes of data quickly and efficiently. 10 lines of pig= 200 lines of Java. Pig provides an ad-hoc way of generating and executing map-reduce jobs on very large data sets. It is designed for rapid development. Java is not needed for Pig. Pig is important because companies like Google, Amazon, Yahoo and Microsoft are collecting various data sets in the form of click streams, search logs and web crawls. Two components are used in Apache Pig:-

1) Pig Latin :- This is a language used to write scripts.

2) Run Time Environment:- It includes parser, optimizer, compiler and execution engine.

Where should use PIG

➔ When there are time sensitive data loads.

➔ When there is need for processing many data sources.

➔ When there is need of analytic insight through sampling.

➔ It is similar to SQL query where the user specifies the “What” and leaves “How” to the underlying processing engine.

Where Not to use PIG

➔ When there is really filthy data formats or completely unstructured data(audio, video, raw human readable text) .

➔ Pig is very slow as compared to MapReduce jobs.

➔ When you would like more power to optimize your code.

V. DEVELOPMENT OF TECHNOLOGIES YEAR WISE

HADOOP

➔ Hadoop was created by persons named Doug Cutting and Mike Cafarella in 2006, when they were working at Yahoo.

➔ It was originally developed to support distribution for the NUTCH search engine project . But in 2002, they felt that nutch architecture can’t be scaled to handle large data sets.

➔ In 2003, Google developed its file system called google file System(GFS).

➔ In 2004, White papers on MapReduce were introduced by Google i.e Initial versions of HDFS and MapReduce were introduced

➔ In 2005, Nutch ported to the new framework. NDFS (Nutch distributed File system) was used with MapReduce.

➔ In January 2006. Doug Cutting joins yahoo.

➔ In februaury 2006, Apache Hadoop project officially started to support the standalone development of MapReduce and HDFS.

➔ Now Hadoop is used to store and distribute very large data sets across hundreds of inexpensive servers that operate in parallel .

➔ Current version of Hadoop is 3.0.1.

NOSQL

➔ The term NOSQL was coined by Carlo Strozz in the year 1998, used this name with his open source, light weight database which did not have an SQL interface.

➔ In the early 2009, with the demand of distributed databases, NOSQL has an unprecedented growth.

➔ NOSQL is used to store, analyze and querying huge amount of unstructured data.

PIG

➔ PIG is a procedural language platform used to develop script for Mapreduce operations, developed by Apache Software foundation and yahoo research.

➔ Its first release come out in September 2008 to perform analysis and processing of data stored in HDFS.

➔ Its stable release is on June 2017.

➔ Its latest version in 0.17.0.

PRESTO

- ➔ PRESTO is developed by facebook in 2012 and released as open source by Apache in 2013.
- ➔ It is written in Java.
- ➔ Its stable release in March 2018.
- ➔ Its latest version in 0.5.7.
- ➔ It is used to develop database query engine using standard SQL.

SQOOP

- ➔ SQOOP is developed by Apache Software foundation.
- ➔ Its initial release was on 2012.
- ➔ Its stable release is on May, 2015.
- ➔ It is written in Java.
- ➔ Its latest version is 1.4.7.
- ➔ It is used to manage large amount of data by importing and exporting data to and from between HDFS and RDBMS.

SPARK

- ➔ SPARK was originally developed at the University of California, Berkeley's AMPLab, later denoted to Apache Software Foundation.
- ➔ Its initial release was on 2014.
- ➔ It is written in Java, Scala, Python and R.
- ➔ Its Stable release is on Feb,2018.
- ➔ Its latest version is 2.3.0.
- ➔ It is used for analysis of huge amount of structured data.
- ➔ SPARK provides an interface for programming entire clusters with data parallelism and fault tolerance.

HIVE

- ➔ HIVE is a community developed project which involves multiple contributors.
- ➔ Its initial release was on 2015.
- ➔ Its stable release is on July 2017.
- ➔ It is written in Java.
- ➔ Its latest version is 2.3.0.
- ➔ It is used for data summarization, query and analysis of data.

VI. CONCLUSION

Big data technologies are gaining considerable attention due to its potential to transform data mining and business analytics practices and the possibility for a diverse range of highly effective decision making tools and services. There are many technologies available for big data. Any enterprise can use these technologies according to their requirements. The technologies highlighted in this paper not only provide smarter decision making but also provide faster time to value. Huge volume of structured, semi-structured and unstructured data getting accrued every day. Analyzing these kinds of data is becoming very helpful for making intelligent business decisions. Big data technologies makes it possible to analyze all available data. With new tools, technologies and infrastructure available at our disposal, it has become much easier to acquire, store and analyze structured, semi-structured and unstructured data in the enterprise. Hadoop is becoming one of the important technologies for enterprise big data platform. Some data are processed real time and some data are processed in batches. There are several challenges for a enterprise to handle huge volume of data like storing, integrating and linking of data, real time processing of data, maintain latency between the time data is generated and the time data is available for consumption, flexibility in data-in/data-out etc. Latest survey by various analysts predict that 80% of the customers are interesting in hadoop and 60% believed hadoop is delivering business value. Top hadoop vendors are doubling their enterprise licenses from the previous years. Data generated in past two years is more than the previous history in total. By 2020, total digital data will groom to 44 zettabytes approximately. By 2020, about 1.7 MB of new information will be generated every second for every person.

VII. REFERENCES

[1] Viktor Mayer-Schonberger, Kenneth Cukier,Big Data: A Revolution that will Transform How we live, work and Think, 2103.

[2] A. Reeve, Managing Data in Motion: Data Integration Best Practice Techniques and Technologies, Morgan Kaufmann, 2013.

[3] X. Dong, D. Srivastava, Big Data Integration, in : Data Engineering (ICDE), 2013 IEEE 29th International Conference on, 2013, pp- 1245-1248.

[4] Knulst " De stand van Hadoop", Incentro, 2012.

[5] Vinayak Borkar, Michael J. Carey, Chen Li, "Inside "Big Data Management": Ogres, Onions, or Parfaits?", EDBT/ICDT 2012 Joint Conference Berlin, Germany, 2012

[6] Economist Intelligence Unit: The Deciding Factor: Big Data and Decision Making. In: Capgemini Reports, pp. 1-24 (2012).

[7] P. Zikopoulos, C. Eaton, Understanding Big Data: Analytics for Enterprise class Hadoop and Streaming data, McGraw-Hill Education, 2011.

[8] I.O'Reilly Media, Big Data Now: 2014 Edition, O'Reilly Media, 2014.

[9] P. Hitzler, K. Janowicz, Linked data, Big data, and the 4th paradigm, Semant. Web (2013) 233-235.

[10] T. White, Hadoop: The Definitive Guide, first ed., O'Reilly Media, Inc., 2009.

[11] D. Borthakur, The Hadoop Distributed File System: Architecture and design, The Apache Software Foundation. (2007) 1-14.

[12] K. Shvschko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), MSST '10, IEEE Computer Society, Washington, DC,USA,2010,pp.1-10

[13] G. Turkington, Hadoop Beginners Guide, Packt Publishing, Limited, 2013.

[14] J. Urbani, S.Kotoulas, J. Maassen, F. Van Harmelen, H. Bal, Webpie: A web-scale parallel interface engine using mapreduce, Web Semant. 10 (2012) 59-75.

[15] A. Thusoo, J.S. Sarma, N. Jain, Z. Shao, P. Chakka, N.Zhang, S. Antony, H. Liu, R. Murthy, Hive- a petabyte scale data

warehouse using Hadoop, in: ICDE '10: Proceedings of the 26th International Conference on Data Engineering, IEEE, 2010, pp. 996-1005.

[16] Learning Presto DB: 2016 Edition, O'Reilly Media.

[17] Dayong Du, Apache Hive Essentials, Packt Publishing, 2015.

[18] Hanish Bansal, Saurabh Chauhan, Shrey Mehrotra, Apache Hive Cookbook, packt Publishing, 2016.

[19] Instants Apache Hive Essentials How-to, Darren Lee, packt Publishing, Limited, 2013.

[20] Balaswamy vaddeman,Beginning Apache Pig,Apress Publications, 2016.

[21] Katbleen Ting and Jarek Jarcec Cecbo, Apache Sqoop Cookbook, Edition 2013, O'Reilly Media.

[22] Holden Karau, Krishan Sankar, Fast Data Processing With Spark 2, 3rd Edition, Packt Publications, 2017.

[23] Tom White, Hadoop: The Definitive Guide.

[24] Khaled Tannir, Optimizing Hadoop for MapReduce, Packt Publishing, 2014.

[25] Jonathan R. Owens, Brain Famiano, Jon Lentz, Hadoop Real World Solutions Cookbook, Packt Publications, 2013.

[26] Dr. Michael Crabb, Robert Gordon University Aberdeen, The Beginners Guide to NOSQL.

[27] Gaurav Vaish, Getting Started with NOSQL, Packt Publications, 2103.

[28] Wayne Chan, Dave Wang, Apache Spark Analytics Made Simple, Databricks Community Edition, 2016.

[29] Rajanarayanan Thottuvaikkatumana, Spark 2.0 for Beginners, Packt Publications, 2016.