Literature Survey and Technology Trends on Hadoop

Varsha Reddy Yasa, Jaynab Khatun and Nabanita Paul
Department of Computer Science & Engineering
University of South Florida, Tampa

***Abstract*— The Usage of immense amount of data has become the most considerable issue in today's business organizations. The evolution of Big Data made a drastic change in the IT technology and was immensely useful for solving the large data problems that were faced in todays world. In this paper we focus on Hadoop which became the leading one for performing most of the operations related to Big Data. This paper deals with the Hadoop Architecture, HDFS/ Hadoop Distributed File System, Map Reduce and few Programming Languages used in here. We also discuss about two important Hadoop Distributions i.e Cloudera and Hortonworks. At last we shown the Technology Trends in Hadoop and deal with Hadoop 3.1.0 which is the trending one.**

Keywords- Big Data, Hadoop, Yarn, Hortonworks, Cloudera, Clusters, Apache 3.1.0

I. Introduction

Big Data deals with huge amounts of datasets where we try to store and analyze petabytes of data. The data can be structured or unstructured data. Big Data is generally described with 3vs: Volume of the data, Variety of Data, and Velocity needed for processing data. The data needed here can be taken from any corner and this will be incorporated with huge analytics, finally used for carrying out the work - related tasks. Here we have to efficiently deal with larger datasets and this can be done by using the conventional or the traditional methods. Due to the advancements in smart technology there was an extreme usage of mobile phones and very huge amounts of data is being transmitted per second. The usage of traditional methods here will not be helpful in handling such data therefore the Big Data comes into picture where we handle such data. The challenges issues for the data will be management. The open source framework Hadoop will be able to handle such kind of applications. Hadoop can deal with massive amount of data and across many servers.

II. Apache Hadoop

The Apache Hadoop project is being endowed by the Apache Software Foundation. The Hadoop is generally based on java. The main goal for the Apache Hadoop is to improve the substantial data. The framework for the Apache Hadoop will help us for handling the distributed data. The data is generally obtained from the Google File System/ GFS. This is being established on the programming standard of the Google Map Reduce. This is basically an open source software. This is highly useful for storing and dealing with the very huge amounts of data. By the usage of different components in the Apache Hadoop like Algorithms for Map Reduce, Different data clusters, distributed processing-various complex problems related to data are being solved. This finally made the most useful technology for the processing of BigData. The services provided are: Data-Accessing, Data-Storage, Data-Altering, Data-Control, and Data-Security. The environment of the Apache-Hadoop contains the Hadoop Kernel, HDFS/ Hadoop Distributed File System, Map-Reduce, and contains other segments such as: Pig, Hive, Zookeeper, Base where the MapReduce and HDFS focusses on data-Storage and data-processing.
The most important attributes of Hadoop are:

- **Scalability:** Here we can combine all the new nodes and this doesnt depend on the data formats.
- **Flexibility:** The two types of data i.e structured, unstructured is generally taken from various sources.
- **Profitability:** Here the output which we get for the parallel computing will be economical.
- **Fault Tolerance:** The loss of node will not stop the operation. Here the task gets redirected to some other point.

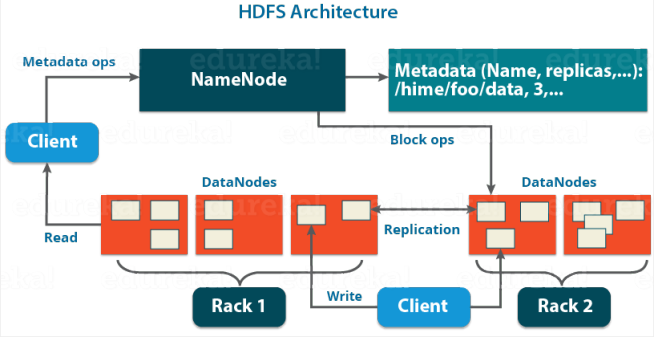

Fig. 1. The Architecture of Hadoop

The Hadoop is one of the distributed file systems. This is basically called as the Hadoop-Distributed File System/HDFS [25]. Here the file system is being made for the purpose of making it run on various number of clusters. This is a commodity hardware. The throughput for the Hadoop-Distributed File System is very high. This is basically Fault tolerant. Therefore, the HDFS is useful for the storage of very large amount of data sets which are taken from various applications. The major advantage for the HDFS is the write once, read many times when it is being compared with the other file systems. There is a division of HDFS to one or

more blocks. This basically depends on the File-Size. The default size for each block will be 64MB. As the design of the HDFS is made to be fault tolerant- The achievement of this is due to the replication of the blocks which are in the HDFS. Whenever there is a data-loss in one of the blocks, we can replace that data which comes from the replication blocks.

The HDFS follows the Master-Slave Architecture. The two main parts of the Architecture will be:

- **Name Node (Master):**
  Here each and every cluster will contain one single Name Node. This name node will basically act like the master of the cluster. The responsibility of the name node is managing the file system-name space, It should respond when a request is made for the data blocks by the client. The Name Node server will help us for managing File System-Name Space. This also helps in controlling the access which is made to files. The Architecture of HDFS is designed such that the data will never be residing on Name Node. Here only the data Node is used for residing of data.
- **Functions of the Name Node:**
  - The Name Node is used for maintaining and managing of the Data Nodes.
  - It is useful for recording the metadata from all the files which are stored in single cluster. The files which are being associated with metadata are:
    FsImage: has the namespace of the file system from beginning of name node.
    EditLogs: the modification that are being made to file system are contained here.
  - It is useful for recording every change that is made to the metadata of the file system.
  - It is used for receiving the block report and also the heart beat from the different data nodes that are present in the cluster and this will make sure that all the data nodes are being alive.
  - It is used for tracking all the records which are present in HDFS blocks and also the nodes in which the blocks are present.
  - It takes the responsibility of the replication factor.
  - Whenever there is a failure in the data node we have to choose for the new replicas of the data nodes.
- **Data Node (Slaves):**
  Here the Data Nodes will are basically the slaves. The role of the Data Nodes is for the storage and retrieval of the data. This storage and retrieval is dependent on the name node request and the client. After the request is being processed the reporting of the request is to be done to Name Node. The data node when compared with the name node makes it non-expensive and does not have good quality and availability.
- **Functions of the Data Node:**
  - These are basically the slaves which are being run on the slave machine.
  - The real data is being stored in the Data Nodes.
  - The purpose of the data nodes is for performing the requests made for write and read from the clients File-System.
  - It is used for sending the heart beats to name node in a periodic manner and also will report the complete health of the file system. The frequency for sending this information is generally set to 3 seconds.
- **Secondary Name Node [24]:**
  The Secondary node will start working with the existing primary name node. This generally helps the name node which is actually present. This is not as backup for name node.
- **Functions of the Secondary Name Node:**
  - The Secondary Name Node is used for reading the files in a constant manner. It is also used for reading the metadata which comes from the Name Node-RAM and then helps to write to the hard disk.
  - It combines the Fsimage and Editlogs by using the name node.
  - The Editlogs will be downloaded by using the name node in some particular intervals and this is applied for the Fsimage. Later the obtained Fsimage will be copied to name node when it is being started.
  - Therfore this secondary name node is responsible for performing the checkpoints in the Hadoop Distributed File System and it is also knowm as the Checkpoint node.
    The Name Node is uniquely identified by the Id number. It is also identified by the data node where the block is being available. Here is data is called as the Metadata. Meta data is basically defined as data about data. The data which is available in the name Node is of very high priority and it is persistent. Whenever there is a data loss it is going to be a permanent one. Therefore a new thing was introduced called as Secondary Name Node. Here it will contain image copy for the Name Node. This is called as the fs-image file. The Secondary Name Node is very useful and this will contain the edit logs. The edit logs will act as log book for all the transactions. The communication which is made in between the Name-Nodes and Data-Nodes is being improved by the Heart-Beats. Here the improvement will be made by the Data Node which will trigger the Name Node Heart beat and is made for every 3 seconds. This will basically contain a report block and also the various blocks which are in Data Node. This is based on name node which will confirm a distinguishable data node and should be active in that moment. Whenever there is a failure of receiving heart beat from a particular data node, there will be a transformation of data from that particular node to some other node. This is generally based on the Metadata.

- **Map Reduce**
  The Map reduce was first introduced by Google and it is model used for parallel processing. Here very huge amounts of data is being processed. There is a distributed task which is taken from the task tracker and this is being assigned to some of the multiple nodes. Now the data is being processed in a parallel fashion. Here the data computation is being done using two basic functions namely Map and Reduce. Here we work with the key/value pairs. The outputs will be the key/value pairs. Here the key/value pair is generally given as the input for the mapper function. This will then produce an intermediate key/ value pair. The output for the intermediate key/value pair is given as input for the reducer function and this will generate the output finally.

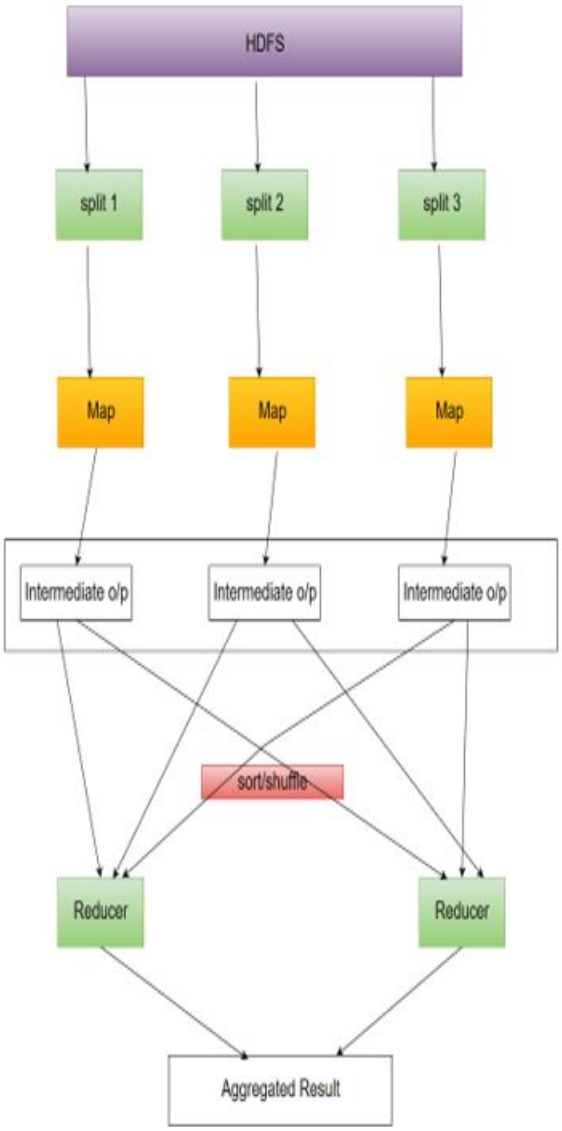

Fig. 2. Map Reduce

There are various phases which happens in Map Reduce. They are:

1) Input Reader Phase
2) Mapping Phase
3) Record Reader
4) Shuffle Phase
5) Sort Phase
6) Reducer Phase

Map Reduce Program

- **Input Reader Phase :**
  In the Map Reduce the initial phase will be the Input reader phase and here the input will be the data from the file and the output is generated in the form of key/value pairs. In the Input reader phase the data is generally made into splits and this is given as input for the mapping phase.
- **Mapping Phase:**
  The Mapping phase will be the next one and here the generated key/value pairs will be the input. The map function in the Mapping phase will help in processing the pairs which were generated. This will help in generating the intermediate key/value pairs. These are finally stored in the local disk.
- **Record Reader:**
  The Record Reader is used for reading the input which is taken from one input split. This is then divided into key/value pairs. These are then passed to the map function. Then it is given to shuffle phase.
- **Shuffle Phase:**
  In the shuffle phase all the intermediate values of the key/value pairs are combined using various mapping functions. This is basically the third phase used in the data flow. Then comes the next phase which is the sort phase.
- **Sort Phase:**
  In the sort phase all the data which is shuffled in the previous phase is being sorted and this sorting is done based on the key. The data here which contains the same key value is being combined. The result which we get from the sorting phase is being given as input to the reduce phase and this will be the final one.
- **Reducer Phase:**
  The Aggregation Operation happens in the Reducer Phase. The aggregation operation is being done by the reducer function. This is also called as the summation-operation. The Remote procedure calls are used by the reducer function which helps in reading the entire buffer. Here the reducer will make sure that it will pass every key with their respective set of values and it is passed to the reducer function which is user defined and it finally give the computed aggregate result. The final result is then stored in the Hadoop Distributed File System.

III. PIG

Apache Pig is a high scripting language for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. Pig is complete platform for data manipulations. Through User Defined Function(UDF) facility in Pig, it allows to invoke code in many language like JRuby, Jython, Java. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.
Pig is very easy to program. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain. Pig is an open source project that is intended to support ad-hoc analysis of very large data, motivated by Sawzall [1], a scripting language for Googles MapReduce[2].
The Pig engine provides a high-level language over the low level map and reduce primitives to simplify programming [3]. To analyze data using Apache Pig, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. There are many reason why programmer need Apache Pig, basically who are not ate Java programming language. Apache Pig is helpful for them while performing any MapReduce tasks. It uses multi-query approach which reduce the length of codes. Apache Pig provides many built-in operators to support data operations like joins, sort, filters, ordering and it also provide nested data types like tuples, bags and maps that are missing from MapReduce.
Apache Pig has many features like rich set of operators, ease of programming, optimization opportunities, Extensibility, UDFs, handles all kinds of data. It supports both structured and unstructured data and store the result in HDFS.
The architecture of Apache Pig is shown in Fig 3. As shown in the figure, there are various components in the Apache Pig Fig-1 framework. Some major components are Parser, Optimizer, Compiler, Execution engine.
Parser: The Parser handles the Pig Scripts and checks the syntax of the script. It includes type checking with other checks. Therefore, an output of the parser will be a Directed Graph. However, it represents the Pig Latin statements and logical operators. In the DAG, the logical operators of the script are represented as the nodes and the data flows are represented as edges.
Optimizer: The logical plan (DAG) is passed to the logical optimizer, which carries out the logical optimizations such as projection and pushdown.
Compiler: The compiler compiles the optimized logical plan into a series of MapReduce jobs.
Execution engine: Finally the MapReduce jobs are submitted to Hadoop in a sorted order. Finally, these MapReduce jobs are executed on Hadoop producing the desired results.
The data model of Pig Latin is fully nested and it allows complex non-atomic datatypes such as map and . Fig-4 is the diagrammatic representation of Pig Latins data model.

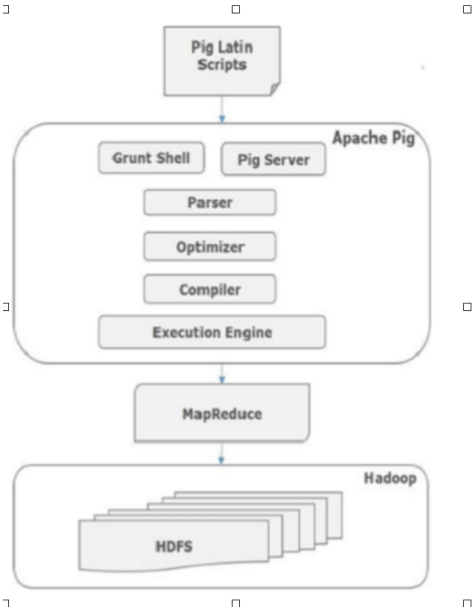

Fig. 3. Architecture of Pig

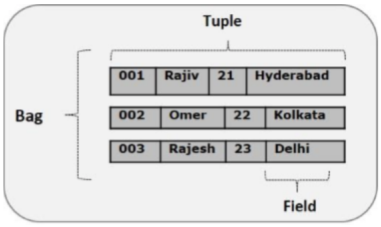

Fig. 4. Diagrammatic representation of Pig Latins data model

Apache Pig has two execution modes 1)Local mode and HDFS mode. In Local mode all the files are installed and run from local host and local file system. In MapReuce mode, the load or process the data in the Haoop File System(HDFS) using apache Pig and Pig Latin statement execute to process the data, a MapReduce job is invoked in the back-end to perform a particular operation on the data that in the HDFS. Apache Pig scripts can be executed in three ways - Interactive mode(Grunt shell), Batch mode(Script), Embedded mode(UDF).

IV. HIVE

The Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. It provides massive scale out and fault tolerance capabilities for data storage and processing on community hardware. Hive an open source data warehouse solution built on top of Hadoop. It supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into map-reduce jobs executed on Hadoop [4]. Experienced database users are often surprised to hear that theres no Hive storage format. In fact, Hive was designed to be completely format agnostic, and accomplishes this feat by supporting pluggable storage format handlers called SerDes

which stands for Serialize/Deserialize.
In Carl Steinbachs word Hive is One of the major value propositions for Apache Hive* is that it can act like an adapter between the
Hadoop* platform and the large ecosystem of data analysis tools that are designed to run on top of relational databases[5].
Hive MetaStore is a system catalog which contains metadata about tables stored in Hive and with SerDes it allows users to cleanly decouple their SQL queries from the physical layers of the data that they are processing. The MetaStore has several different types of catalog metadata like table-to-column, table-to-storage-location, and Table -to-SerDes.
Although Apache Hive some limitations. Hive cannot perform low-level insert, updates or deletes. So it cannot be used for online transaction processing(OLTP). Hive is high latency and high throughput system.
The architecture of Apache Hive has shown Fig-3, where round rectangles are components in Hive and standard rounded rectangles are advanced components. Hive focuses on two interfaces to users to load their queries. These interfaces are Command Line Interfaces (CLI) and HiveServer2 [6]. The major components of Hive architecture as figure Fig-2 are External Interfaces, Thrift Server, Metastore, and Driver.

Command Line Interface(CLI): CLI is default is shell where users can execute hive Queries and Command directly.

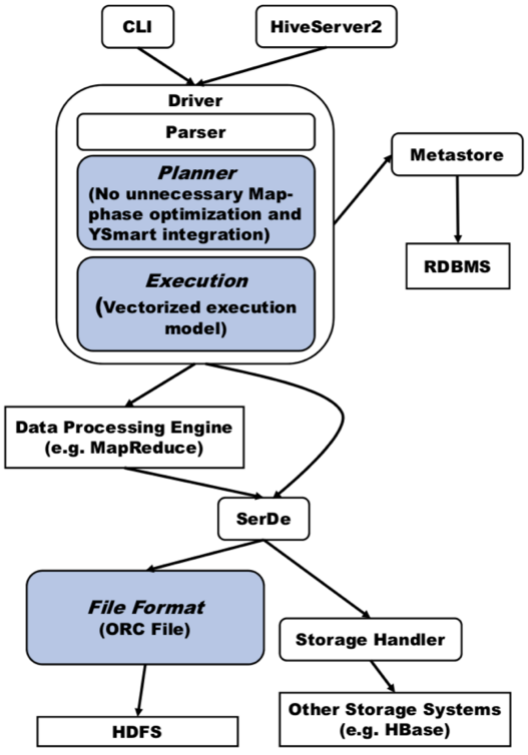

Fig. 5. Architecture of Hive

Web Interface: Web interfaces usually use for UI, appli- cation programming interfaces (API) like JDBC and ODBC. It provides different types data types like Numeric, Date/Time, String Miscellaneous and operators like Relational, Arithmetic, Logical, String operator on complex Types etc.

Thrift Server: The Hive thrift Server allows client API to execute HiveSQL statements. Thrift [7] framework is for cross-functional language where a server written in one language and can also support in other languages.

Driver: Hive Driver is use for receiving the queries submitted by Thrift server, JDBC, ODBC, CLI, Web Interfaces. It manages the Hive query language statements during compilation, optimization and execution. The Driver invokes the compiler with the HiveQL strings. The Compiler converts the string plan which consists of DDL statement and HDFS operation.

HiveServer2 is a server interface which performs basic functions. It enables the remote clients to execute queries with Hive and it helps to retrieve the results of mentioned queries. There are some advanced features in HiveServer2 is based on Thrift RPC like Multi-client concurrency and authentication.Queries used for data retrieval and processing are analyzed by the Query Planner. Hive translates queries to executable jobs for an underlying data processing engine that is currently Hadoop MapReduce [8]. For a submitted query, the query planner walks the AST of this query and assembles the operator tree to represent data operations of this query [9].

V. SQOOP

Apache Sqoop transfer bulk data between apache Hadoop and Structured data-stores like relational databases, enterprise data warehouses, NoSQL system. Sqoop allows easy imports and exports of data sets between databases and HDFS. Sqoop occupies a place in the Hadoop ecosystem to provide feasible interaction between relation database server and Hadoops HDFS. It is used to import data from relational databases such as MySQL, Oracle, DB2 to Hadoop HDFS and export from Hadoop file system to structured relational databases.

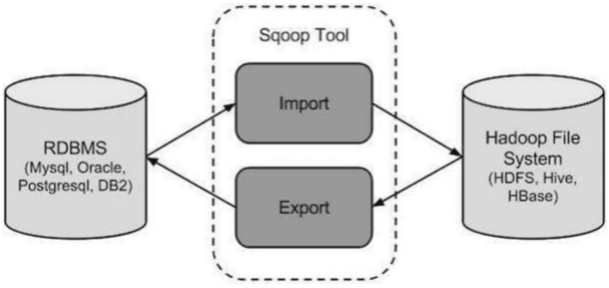

Fig. 6. Architecture of Sqoop

The main goal of Sqoop is to enable analytics frameworks to benefit from the computational resources of an object

store for optimizing job execution in disaggregated clusters [10]. This is where Apache Sqoop fits in. Apache Sqoop is currently undergoing incubation at Apache Software Foundation [11]. The architecture of Apache Sqoop is shown Fig-6. The main two part of Apache Sqoop is Importing data and Exporting data.

Sqoop import: the import tools imports every tables from relational databases and saves the imported data into Hadoop File system. YARN coordinates data ingest from Apache Sqoop and other services that deliver data into the Enterprise Hadoop cluster.

Import data: The import sub command which instruct Sqoop to initiate import. To import data Sqoop perusals the databases for collecting the necessary metadata then it submits to cluster. Sqoop still enables different data formats for data imports. For instance, Avro data format can be used by the user for easily importing data. Example: avrodatafile with the import command. To perform import operation, Sqoop provides several options like Hbase-create-table which enables Sqoop to create the Hbase Table and Hbase-table which specifies table nam to use [12]. The Figure Fig-7 has shown importing of data from relational databases and store into Hive or Hbase by using the command with import sub-command

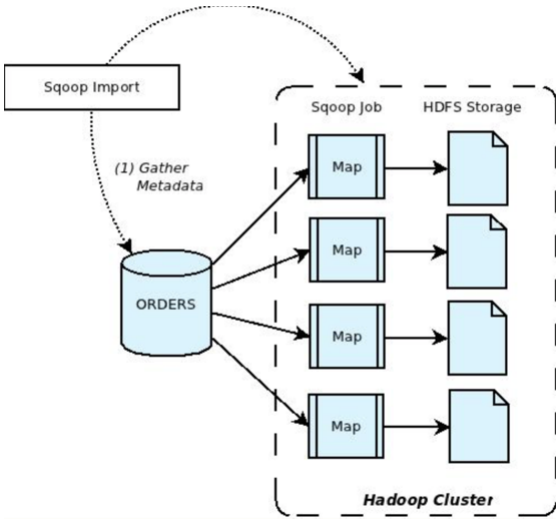

Fig. 7. Importing data in Sqoop

Export data: The Sqoop export tool export data which is a set of files from HDFS back to relational databases. The first step is to examine the database for metadata, followed by the second step of transferring the data. The input dataset is divided into a number of splits using Sqoop, then Sqoop sends those splits into databases using individual map tasks. Each of these map task performs this transfer over many transactions for ensuring minimal utilization and optimal throughput [12]. Export is done in two step which has shown in figure Fig-8.

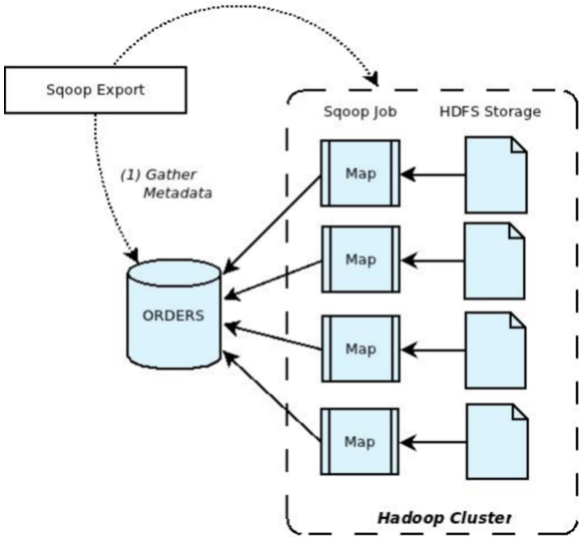

Fig. 8. Exporting data in Sqoop

VI. Flume

Flume [15] is a widely used open-source real-time data processing framework. Flume is a widely used open source framework in industry for real-time big data processing. Table 1 is a list of companies who have publicly mentioned its usage [16]. Apache Flume is service ingestion mechanism for collecting aggregating and transporting large amount of streaming data such as log files, events from many source to centralized data-server. It is highly reliable, distributed and configurable tool. Flume is application built on top of Hadoop which is used for moving of huge amounts of streaming datasets into Hadoop Distributed File System(HDFS)[17]

The architecture of Apache Flume is shown figure Fig-7. In this figure the data generators produce data which is collected by flume Agent running on them. Then Data collector collects all the data from Agents and Combine them into file such as HDFS or Hbase so it can be transferred to the Centralized store. Flume Agents is an independent java deamon process which receives the data from clients os other agents Facebook, Twitter and transfer it to Data collector. Agents has three major components like Source, Channel and Sink. The other additional components of Flume Agents are Interceptor, Channel Selector, Sink Processor.

Fig. 9. Architecture of Flume

Examples of Flume channel types are a JDBC channel, a file channel, and a memory channel [18]. Apache Flume is used in online application where real-time event processing is fundamental. A beautiful instagram can be experience using Apache Flume for any desktop or mobile devices.

VII. Hadoop Distributions

A. Hortonworks Data Platform

Apache Hadoop may be the basis for an incredible and adaptable data refinery. Furthermore, Hadoop being an open source software, this power and adaptability is unreservedly accessible to all. However, for some companies, the intricacy of coordinating the different Hadoop segments with each other, and with existing data designs, speaks to a undisclosed expense and a obstacle to effective selection.

Hortonworks[19] was developed to take out the obstacles to

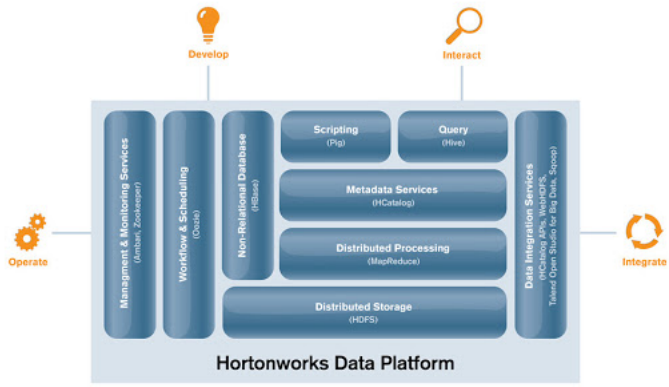

Fig. 10. Pictorial representation of Hortoworks Data Platform [19]

Hadoop appropriation, making the Hadoop platform simpler to utilize and expend, and making its advantages more available to everyone, including end user companies, hardware and software merchants, and consultants and systems assemblers.

To accelerate this objective, Hortonworks offers the Hortonworks Data Platform, a pre-incorporated bundle of fundamental Apache Hadoop segments, which enables users to effectively bridle the capacity of Hadoop and boost the value of all their data.

Hortonworks Data Platform conveys, in a solitary, firmly coordinated bundle, mainstream Apache Hadoop ventures, for example, HDFS, MapReduce, Pig, Hive, HBase and Zookeeper. To this base, Hortonworks Data Platform incorporates extra open source advancements that make the Hadoop stage more reasonable, open, and extensible. An entire arrangement of open APIs is given, making it simpler for undertakings and ISVs to incorporate and expand Apache Hadoop.

Making Hadoop available starts with installation and setup. Normally a relentless task,installation and setup of Hadoop is made even more intricate by the way that the open source software packages that make up the Hadoop platform are autonomously created and regularly refreshed codebases.Moreover, each of these packages have their own release schedules, versions and dependencies.

To guarantee a reliable and stable platform for enterprise utilize, Hortonworks Data Platform incorporates just stable segment versions that have been completely coordinated, tested and affirmed as a major aspect of Hortonworks' broad Q/A procedure, and are bolstered by the organization's multi-year support and maintenance policy.

Hortonworks Data Platform provides installation and configuration tools that are easy to install, deploy and manage. The Hortonworks Management Center was developed from Apache Ambari, an open source installation, configuration and management system for Hadoop, and is incorporated into Hortonworks Data Platform. The Hortonworks Management Center furnishes an extensive web dashboard that coordinates monitoring, metrics and alerting data into a consolidated Hadoop-specific management console.

Imperative metadata management capability is incorporated into Hortonworks Data Platform by means of an open source software called Apache HCatalog. HCatalog furnishes intensive metadata services like table and schema management, to the majority of the platform constituents. Furthermore, it gives a strategy to more profound collaboration with third-party data management and analysis tools, enhancing interoperability.

Beyond innovation, as the enterprise driving distribution of Apache Hadoop, Hortonworks Data Platform is upheld by an incredible cooperation of partners the consist of pioneer software and hardware vendors, and frameworks integrators. These organizations help guarantee that your interest in Hadoop broadens and supplements existing IT investments and venture connections.

Hortonworks Data Platform (HDP) is facilitated 100% by Open Source Apache Hadoop. HDP furnishes all the Apache Hadoop-related projects imperative to integrate Hadoop alongside an EDW (Enterprise Data Warehouse) as part of a Modern Data Architecture.

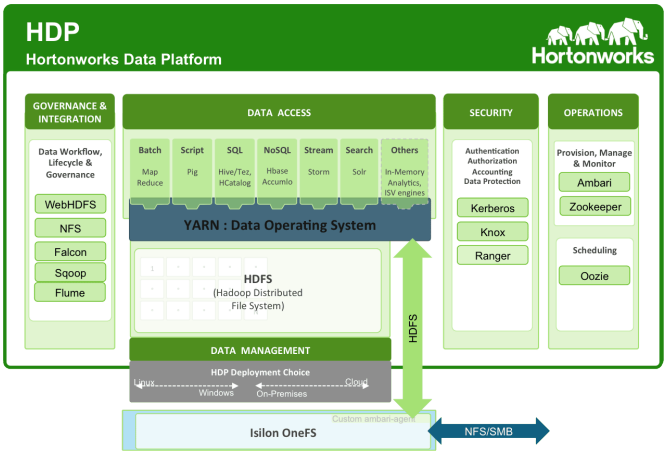

Fig. 11. Isilon-MDA [19]

1) Deployment Options for Hadoop: **On-premises:** HDP is the only Hadoop platform that works across Linux and Windows.

Cloud: HDP can be executed as part of IaaS, and also fuels Rackspaces Big Data Cloud, and Microsofts HDInsight Service, CSC and many others.
Appliance: HDP can also execute on commodity hardware by default as well as be purchased as an appliance from Teradata.

2) The Attunity Solution for Hadoop : Attunity[20] conveys a superior performance solution that migrates Big Data into and out of Hadoop without breaking a sweat. Attunity Replicate facilitates access and stack gigantic volumes of data for investigation in the cloud and datacenter. What's more, Attunity Maestro arranges and computerizes data transmission and deployment procedures of Big Data, applications, and extensive document resources. The At-

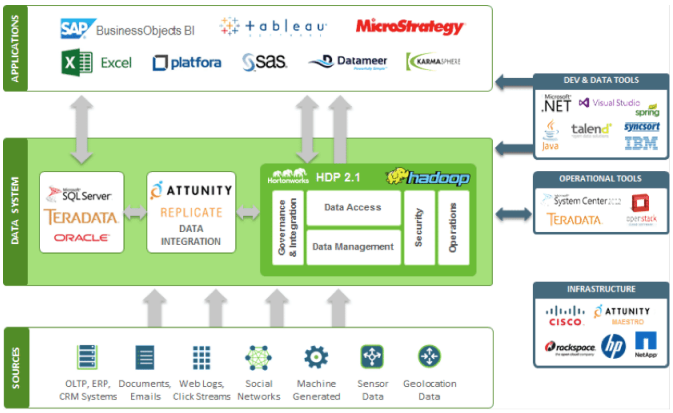

Fig. 12. Attunity-Hortonworks [20]

tunity equips high-speed availability to gathering data out of most enterprise information sources. This data is then consequently stored into Hadoop where it is made accessible for any Hadoop application to chip away at. Users who are entrusted with bringing data into Hadoop utilizing Attunity Replicate don't have to learn Hadoop to play out this activity. This decreases the requirement for extra effort for Hadoop training - amplifying its maximum capacity by taking advantage of any data source and destination and additionally maintaining a strategic distance from extra interests in preparing and procuring. The natural and industry proven Attunity Replicate incorporates a GUI that empowers users to quicken Hadoop installations for any data delivery. No expert coding is required for performance improvement on distributed computer systems.
Key Features/Benefits provided by Attunity are:

- High-performance connectivity to Hadoop via native APIs for data comsumption and publication.
- Autonomous schema generation in HCatalog.
- Dragging & dropping of configuration with the help of "Click-2-Replicate" design
- High-speed data load options:
  - Full reload with overwrite
  - Insert only appends
  - Change Data Capture
- Spontaneous data filtering and transformation.
- Compression supported by Gzip
- Supervising dashboard with the help of web-based metrics, alerts and log file management.

Using Attunity's solution for Hadoop, enterprises can:

- Decrease the time and resources needed to migrate data for Hadoop.
- Reduce the costs involved in moving data for Hadoop.
- Migrate data in batch and gradually with low latency.
- Automate mobility between Hadoop and data warehouses.
- Utilize Hadoop as both a source and a destination system.
- Maintain the data supply chain - including data lakes - through a visual user interface.

Big Data has altered the manner in which that we utilize and manage data. Currently, Hortonworks has a larger volume of data than it ever had before, in higher speeds from more sources across the company. Companies cannot stand to lose prospective business because of time spent "data wrangling" with the end goal to dig their data for valuable pieces. Cooperating, Attunity and Hortonworks offer an answer for easing those difficulties. Companies are utilizing the joint solution today to significantly enhance the flow and convenience of Big Data to accomplish quicker time-to-value advantage.

B. Cloudera Hadoop Distribution

Cloudera[21] gives a versatile, adaptable, coordinated platform that makes it simple to oversee quickly expanding volumes and assortments of data in a company. Cloudera solutions empower the user to deploy and oversee Apache Hadoop and related projects, control and examine the data, and keep that data secure.

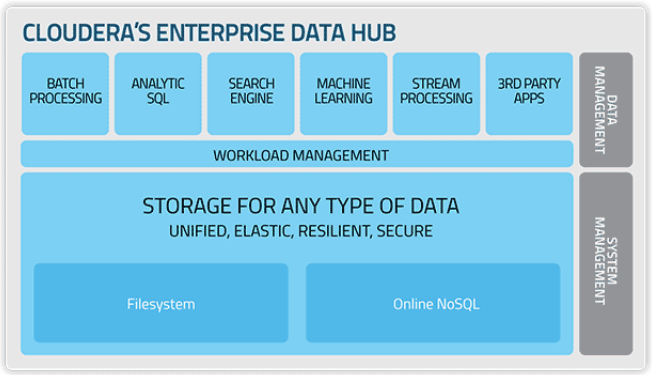

Fig. 13. Cloudera Data hub [21]

Cloudera provides the following products and tools:

- **CDH :**The Cloudera distribution of Apache Hadoop and similar open-source projects that include Impala and Cloudera Search. Along with these, CDH also furnishes security and integration with various hardware and software solutions.

– **Cloudera Impala :** A extremely parallel processing SQL engine for intuitive business analytics and intelligence. Its profoundly improved architecture makes it perfectly suited for conventional BI-style queries with joins, conglomerations, and subqueries. It can query Hadoop data documents from an collection of sources, including those delivered by MapReduce activities or stacked into Hive tables. The YARN resource management segment gives Impala a chance to coincide on clusters running batch tasks simultaneously with Impala SQL queries. A user can maintain Impala along with Hadoop segments through the Cloudera Manager UI, and secure its data through the Sentry authorization framework.
The Impala solution is composed of the following components:

  * **Clients -** Interfaces like Hue, ODBC clients, JDBC clients, and the Impala Shell can all interact with Impala and are most commonly used to issue queries or complete administrative tasks such as connecting to Impala.
  * **Hive Metastore -** Metastores are used to store information regarding the data available to Impala. For instance, the metastore informs Impala which of the databases are accessible and what is the schema of those databases. As a user performs data operations to create, drop, and alter database objects, stack data into tables, etc with the help of Impala SQL explanations, the significant metadata changes are consequently communicated to all Impala hubs by the devoted inventory service added in Impala 1.2.
  * **Impala -** Running on DataNodes,this procedure facilitates in coordination and execution of queries. Each object of Impala can get, plan, and coordinate queries from Impala customers. These queries are conveyed among Impala hubs, and these hubs execute the queries in parallel.
  * **HBase and HDFS -** Provides the storage for data to be queried.

– **Cloudera Search :** Cloudera Search furnishes a real-time access to stored or ingested into Hadoop and HBase. Searches render real-time indexing, batch indexing, full-text exploration and navigated drill-down and in addition a straightforward, full-text interface that requires no SQL or programming expertises. Completely coordinated in the data-processing platform, Search utilizes the adaptable, versatile, and powerful storage framework included with CDH. This wipes out the need to move huge data collections between warehouses to perform business undertakings.
Cloudera Search consists of Apache Solr, which in turn includes Apache Lucene, SolrCloud, Apache Tika, and Solr Cell. Cloudera Search is a component of CDH 5. Using Search with the CDH infrastructure provides:

  * Reduced infrastructure
  * Improved production visibility
  * Faster perception of various data types
  * Faster problem solving
  * Interaction and platform access for more users and use cases.
  * Improved search services with Scalability, flexibility, and reliability on the same platform used to run other workloads on the same data

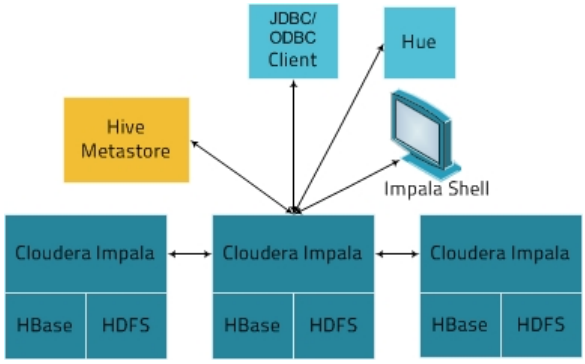

Fig. 14. Impala Architecture [21]

- **Cloudera Manager:**Cloudera Manager is an advanced application used to install, oversee, monitor, and determine issues with CDH organizations. Cloudera Manager furnishes the Admin Console, an web UI that makes organization of a company's data simple and straightforward. It additionally incorporates the Cloudera Manager API, which can be used to acquire cluster health data and measurements, and configure Cloudera Manager.
- **Cloudera Navigator:** Being overall data security and management system for CDH, Cloudera Navigator empowers data administrators and scientists to investigate the data storage in Hadoop, while keeping the storage and management of encryption keys simple. Cloudera Navigator enable companies to abide to stringent Compliance and administrative requirements by with the help of vigorous auditing, data management, lifecycle management, and encryption key management.

C. Hortonworks and Cloudera merger

On October 3, 2018, Cloudera, Inc. also, Hortonworks, Inc.jointly declared that they have gone into a complete understanding under which the organizations will merge in an all-stock merger of equals. The exchange, which has been unanimously affirmed by the Boards of Directors of the two organizations, will make the world's dominant cutting edge data platform supplier, spreading over multi-cloud, on-premises and the Edge. The collaboration sets up the business standard for hybrid cloud data management, quickening client appropriation, network improvement and partner engagement.

This merger will bring the following changes to the Hadoop Data Platform paradigm:

- Establish cutting-edge data platform leader with ungraded scale and resources to deliver the industry's first enterprise data cloud, furnishing the comfort of use and flexibility of the public cloud from everywhere starting from the data center to to the Edge.
- Develop a high-standard platform as well as well-defined industry standard from the Edge to AI, significantly facilitating users, partners and the community
- Escalate market improvement and powers innovation in machine learning/AI, streaming, data warehouse, IoT, hybrid cloud
- Extend business opportunity with complementary contributions, including Hortonworks DataFlow and Cloudera Data Science Workbench.
- Enhance associations with open cloud merchants and frameworks integrators.

VIII. Technology Trends in Hadoop: 3.1.0

Apache Hadoop 3.1.0[22] intends to implement functionalities that were absent from Apache Hadoop 2.0. Hadoop 3 combines the efforts of hundreds of contributors over the last five years since Hadoop 2 launched.
The advantages that Hadoop 3.1.0 has over Hadoop 2 are:

- **Agility & Time to Market :** Hadoop 3 containerization improves the use of containers by adding agility and package isolation story of Docker. A container-based service facilitates in building apps fast and release it in minutes. Additionally, it furnishes faster time to market for services.
- **Total Cost of Ownership :** Hadoop 3 address the issue of storage overhead of Hadoop 2. For instance, in Hadoop 2, if there are 6 blocks and 3x replication of each block, the total space required is 18 blocks of space.
  Using erasure coding in Hadoop 3, if there are 6 blocks, the total space required is 9 block space 6 blocks and 3 for parity reducing storage overhead. The end result -instead of the 3x hit on storage, the erasure coding storage method will have an overhead of 1.5x, while furnishing the same kind of data recoverability. It reduced the storage expense of HDFS by 50% along with retention data durability. Storage overhead is therefore decreased from 200% to 50%. This also provide users and enterprises the benefit from the tremendous cost savings.
- **Scalability & Availability :** While, Hadoop 2 and Hadoop 1 only utilized a single NameNode to manage all Namespaces, Hadoop 3 enhances scalability by using multiple Namenodes for multiple namespaces for NameNode Federation.
  Additionally in Hadoop 2, there is only one standby NameNode, on the other hand, Hadoop 3 comes with multiple standby NameNodes. If one standby node stops working over the weekend, other standby NameNodes can be used so that the cluster can continue to operate.

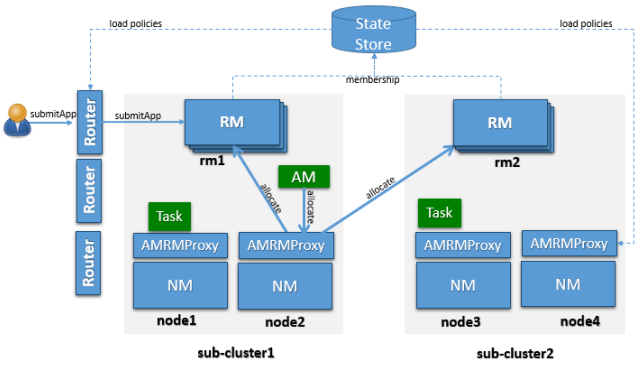

Fig. 15. Hadoop 3.1 Architecture [23]

This functionality provides a longer servicing window. Apart from all these, Hadoop 3 enhances the timeline service v2 and upgrades the scalability and reliability of timeline service.

- **Hybridization :** Hortonworks Data Platform (HDP) 3.0 developed on Apache Hadoop 3.1, is augmented for the cloud, guaranteeing automated cloud provisioning to simplify big data deployments while improving the utilization of cloud resources. The platform incorporates built help for the majority of the real cloud protest stores: Azure Data Lake Store (ADLS), Azure Storage Blob, Amazon S3, and Google Cloud Storage (GCS) technical preview. HDP is cloud-cynic. Clients can utilize Cloudbreak for simple provisioning of HDP clusters to their choice of supplier. Moreover, there are service connectors to cloud including Apache HBase and S3 (specialized see), and Apache Spark with S3Guard for higher inquiry execution.

A. New Use Cases

1) While Hadoop 2 does not provide support for GPUs, Hadoop 3 empowers scheduling of extra resources, for example, disks and GPUs for better collaboration with containers, deep learning & machine learning. This functionality gives the premise to supporting GPUs in Hadoop clusters, which upgrades the performance of calculations required for Data Science and AI use cases.
2) Unlike Hadoop 2, Hadoop 3 has intra-node disk balancing. In the event that a user is adding new storage to a current server with older drives, this prompts unevenly disks space in every server. Using intra-node disk balancing, the space in each disk is uniformly distributed.
3) With Hadoop 3, intra-queue preemption is upgraded by allowing preemption between applications within a single queue. This facilitates the user to prioritize tasks within the queue based on user limits and/or application priority

IX. CONCLUSIONS

Apache Hadoop is highly reliable distributed processing of vast amount of data using few programming language. Big data is analysis is new use case for Hadoop. Big data offers new data formats compatible with Enterprise Data Warehouse, data lake analytics and data offload and consolidation. Apache Hadoop is set of software technology components bind into package which is scalable system optimized for data analysis. Hadoop is open source platform where the key components are Yet Another Resource Negotiator (YARN), Hadoop Distributed File System (HDFS), Apache Pig, Sqoop, Hive, Flume, Map-Reduce engine etc. Apache hive data warehouse software which allows reading, writing, managing large data sets using SQL. Apache Sqoop is used to transfer data between apache Hadoop and structured data sets like Enter Data Warehouses, NoSQL system. The New version of Hadoop is Apache Hadoop 3.1.0 release in December 2017. Apache Hadoop 3.1.0 contains a number of significant features and enhancements like YARN, HDFS, MapReduce. Hortonworks works with IBM Ends to expands Hybrid Open Source for Hadoop Distribution. Github, IBM Cloud are great source of Hadoop Distribution. Future work includes External use of storage system enhance the Engine like MapRduce, support first class GPU and FPGA. The upcoming version of Apache Hadoop 3.1.2 which are in testing phase and downstream adoption phase.

REFERENCES

[1] R. Pike et al. Interpreting the data: Parallel analysis with Sawzall. Scientific Programming, 13(4) :277- 298, 2005.

[2] Parallel Data Processing with MapReduce: A Survey, SIGMOD Record, December 2011 (Vol. 40, No. 4).

[3] Analyzing Massive Astrophysical Datasets: Can Pig/Hadoop or a Relational DBMS Help?, University of Washington, Seattle, WA.

[4] Hive - A Warehousing Solution Over a Map-Reduce Framework, Facebook Data Infrastructure Team.

[5] Apache Hadoop* Community Spotlight Apache Hive, http://trgcqcontent.cps.intel.com/, March 2013.

[6] https://cwiki.apache.org/confluence/display/Hive/Setting+up+HiveServer2.

[7] Apache Thrift.http://incubator.apache.org/thrift.

[8] https://hadoop.apache.org/.

[9] Major Technical Advancements in Apache Hive,The Ohio State University, Hortonworks Inc., Microsoft

[10] Too Big to Eat: Boosting Analytics Data Ingestion from Object Stores with Sqoop, Francesco Pace, Daniele Venzano, Pietro Michiardi Eurecom (France) francesco.pace—daniele.venzano—pietro.michiardi@eurecom.fr

[11] http://incubator.apache.org/sqoop.

[12] An Efficient HADOOP Frameworks SQOOP and Ambari for Big Data Processing, IJIRST International Journal for Innovative Research in Science and Technology— Volume 1 — Issue 10 — March 2015 ISSN (online): 2349-6010

[13] https://blogs.apache.org/sqoop

[14] http://couponmarketer.mobiclients.net

[15] Apache Flume: http://flume.apache.org/

[16] Apache. Flume. Public List of Companies Using Flume. https://cwiki.apache.org/FLUME/powered-by.html

[17] Palanisamy, B., Singh, A., and Liu, L, ost-effective resource provisioning for mapreduce in a cloud, IEEE Transactions on Parallel and Distributed Systems, 2015, pp:1265-1279.

[18] D Vohra - Practical Hadoop Ecosystem, 2016 - Springer, Apache Flume

[19] https://hortonworks.com/wp-content/uploads/2012/06/Apache-Hadoop-Big-Data-Refinery-WP.pdf.

[20] http://hortonworks.com/wp-content/uploads/2012/06/Hortonworks-Attunity-whitepaper.pdf.

[21] https://www.cloudera.com/documentation/enterprise/5-6-x/topics

[22] https://hortonworks.com/blog/announcing-hdp-3-0-faster-smarter-hybrid-data/

[23] https://hadoop.apache.org/docs/r3.1.1/hadoop-yarn/hadoop-yarn-site/Federation.html

[24] https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/

[25] Konstantin Shvachko, et,.al., "The Hadoop Distributed File System", MSST '10 Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, May 03 - 07, 2010 .