Modern Applied Science; Vol. 14, No. 9; 2020
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Big Data Ingestion and Preparation Tools

Jaber Alwidian[1], Sana Abdel Rahman[1], Maram Gnaim[1] & Fatima Al-Taharwah[1]

[1] King Abdullah I School of Graduate Studies and Scientific Research, Princess Sumaya University for Technology (PSUT), Amman, Jordan

Correspondence: Maram Gnaim, King Abdullah I School of Graduate Studies and Scientific Research, Princess Sumaya University for Technology (PSUT), Amman, Jordan

Received: January 25, 2020 Accepted: August 24, 2020 Online Published: August 27, 2020

doi:10.5539/mas.v14n9p12 URL: https://doi.org/10.5539/mas.v14n9p12

Abstract

Developing in Big Data applications become very important in the last few years, many organizations and industries are aware that data analysis is becoming an important factor to be more competitive and discover new trends and insights. Data ingestion and preparation step is the starting point for developing any Big Data project. This paper is a review for some of the most widely used Big Data ingestion and preparation tools, it discusses the main features, advantages and usage for each tool. The purpose of this paper is to help users to select the right ingestion and preparation tool according to their needs and applications' requirements.

Keywords: big data, Hadoop, HDFS, data ingestion, data preparation

1. Introduction

In recent years the data is growing quickly, multiple sources such as computers, social media and mobile phones are generating large volume of data with different format, namely structured, semi-structured and unstructured. (Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S., 2018) (Erraissi, A., Belangour, A., & Tragha, A., 2018)

Big Data require to ingest, clean, process and extract an important value from the data. Different models, hardware's and technologies have been developed for Big Data to provide more trustable and accurate results, most of these technologies are open source and available to handle the volume and variety of data. Hadoop is the popular framework for Big Data that integrate different technologies for ingesting and analyzing the different type of data. However, in different cases it is challenging to choose the best technology to be used as this depend on different parameters such as cost, performance and support. (Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S., 2018) (Mohamed, E., & Hong, Z., 2016)

Data ingestion process is an important step in building any big data project, it is frequently discussed with ETL concept which is extract, transform, and load. Traditionally, ETL was built for moving the data from source to destination via created pipeline, but this process is slow and not time-sensitive. Modern applications aim to provide a model for real time processing and decision making, in this case the ETL is created with different architecture to solve the latency problem and to deal with streaming data such as website clicks, sensors and telecommunications, so the new arrived data will be transferred immediately for processing. (Meehan, J., Aslantas, C., Zdonik, S., Tatbul, N., & Du, J., 2017)

Data ingestion process should be handle the different volume, speed and variety of data. It can be batch data ingestion or Stream data ingestion. This paper discussed the Big Data ingestion process with different tools for batch and stream ingestion such as Sqoop, NIFI, Flume and Kafka. Each tool is discussed with its' features, architecture and real use case. It has a comparison for big data ingestion tools based in different criteria, this comparison will help users to choose the tool that satisfies their needs. Also, it mentioned the data preparation process that aims to clean, validate and reduce the ingested data.it mentioned some tools for data preparation like Hive, Impala, Storm and Spark.

The paper has the following structure, section two introduced the data sources and the types of data. Section three presented the big data ingestion concept, parameters and challenges, it reviewed some of the ingestion tools categorized based on ingestion type either batch or stream, and it discussed details about each tool. Section four introduced the data preparation process which is pre-processing step for data quality enhancement, and mentioned some tools for data preparation with its main characteristics and real use case.

2. Data Source

The volume of data that used in big data projects is very large, also the sources and format of data are changing rapidly. There are two main data sources internal and external, internal sources which are controlled by the organizations and included data about daily operations of the company that collected and stored in databases, in this case we are discussing about structured data, external sources refer to all the data that retrieved from external sources that are not controlled by the organization. (Bucur, C., 2015) (Erraissi, A., Belangour, A., & Tragha, A., 2018)

Big data has different data sources, social media is the most important source, Twitter and Facebook generate very large amount of data such as tweets, profiles and likes, this data can be analyzed and provide important value, for example analysis of social media data that related to new product can provide better understanding about customer satisfaction. Log files are another source of data, for example clicks on specific website can be logged into web log files, and these logs can be analyzed to understand the online user's behavior. Sensors and machines such as medical devices, smart meters and road cameras generate large volume of data and these data can be analyzed and provide valuable output. Geospatial data that generated by cell phones is another source of data that can be used by another application. 0

There are three types of data: (Erraissi, A., Belangour, A., & Tragha, A., 2018)

- Structured data: it refers to the data that has fixed format and stored into rows and columns, such as data that stored into relational databases.
- Unstructured data: it refers to data that does not have specific format or structure and this make it difficult for processing, it can be textual like emails or non-textual like audio and video.
- Semi-Structured: it lies between the above mentioned types, it does not have complex format however it has specific information such as tags, xml file is an example of semi-structured data.

Figure 1. Data Sources (Oughdir, L., Dakkak, A., & Dahdouh, K., 2019)

3. Data Ingestion

Data ingestion is a process of moving and transferring different types of data (structured, un-structured and semi-structured) from their sources to other system for processing, this process starts with prioritizing data sources then validating information and routing data to the correct destination. (MATACUTA, A., & POPA, C., 2018) (Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., & Arocena, P. C., 2019)

3.1 Data Ingestion Parameters

To complete the data ingestion process effectively we should use the correct data ingestion tool that is compatible with the business case, many parameters should be studied when choosing the correct data ingestion tool. Data size is one of the data ingestion parameters that refers to the large amount of data that generated by different sources and needs to be ingested. Data format is another parameter that refer to different format of data, it can be structured like tabular or unstructured like images or semi-structured like xml files. Data frequency is an important parameter of data ingestion which refers to batch or real time, in real time the data is processed once it received but in the batch case the data is stored in batches and transferred in specific time interval. Data velocity refers to the speed of flow of data from different sources, so the data ingestion pipeline should be compatible with the business data traffic as sometimes the traffic is high or low. (MATACUTA, A., & POPA, C., 2018)

3.2 Data Ingestion Challenges

The number of smart and IOT devices are increasing rapidly, so the volume and format of the generated data are increasing and this will be considered as the biggest challenge of big data ingestion as the business needs to read the large volume of generated data in acceptable speed. When data is coming from different sources it will be in different format and structure, it should be transformed into one common format and this need complicated and time consumed processing. The produced data is changing rapidly and these modifications should be ingested in order to update the original data, this is complicated process because different format of data maybe involved in the ingestion and low latency in data ingestion is needed by business. Data ingestion process is restricted to security and compliance laws and this will make it complex, costly and time consuming. (Shahin, D., Hannen Ennab, R. S., & Alwidian, J., 2019)

3.3 Batch Data Ingestion

The data is ingested in batches every defined interval of times such as ingesting all the transaction data from specific day at the end of the day, or when the data reaches a certain size, batch data ingestion is useful for offline analytics that aren't time-sensitive.

3.3.1 Sqoop Apache

Sqoop is a tool for transferring bulk of data between Hadoop and relational databases or mainframes. As shown on Figure2 below, sqoop can import the data from different types of relational database management systems (RDBMS) like Oracle, MySQL or a mainframe to Hadoop distributed file system (HDFS), then process the data using Hadoop MapReduce and export the processed data to RDBMS. (Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., … Zaharia, M., 2015)

Sqoop is connected to the RDBMS by using JDBC connector and depends on the RDBMS schema for the imported data. Sqoop depends on MapReduce which provides the parallel processing functionality, so the output of Sqoop import process will be multiple files in HDFS —delimited text files, binary Avro or Sequence Files— containing the imported data.

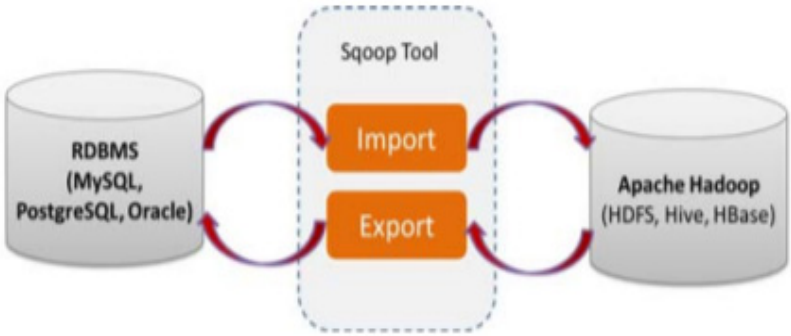

Figure 2. Sqoop Functionality (Cheng, Y., Zhang, Q., & Ye, Z., 2019)

Sqoop is used by many applications that related to different sectors, it was used by "Coupons.com" which is online marketer in order to transfer the data between IBM -Netezza and Hadoop, also it was used by" Apollo Group" which is educational company to import and export data between Hadoop and relational databases. (Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., … Zaharia, M., 2015)

Sqoop also used within different researches, it was used by electronic medical records data analysis on cloud, this research is about analyzing health care and electronic medical records (EMR) data, the purpose of this research is to help the health care organizations to detect any un-usual measurements that need immediate action and support the decision making process. Sqoop is used to import bulk of electronic medical records from the related database and insert the data into hive table, then analyzing the data by using MapReduce algorithms and finally export the data again to external database on cloud. (Rallapalli, S., & Gondkar, R. R., 2015)

Sqoop is also used within crime data analysis research, the purpose of this analysis is to analyze the population, crimes and crime rates and this is very critical issue for the governments in order to make strategic decisions to apply the law and to keep the citizens safe from the crime. The related data is loaded from RDBMS to HDFS by using apache Sqoop and apache Flume was used to load unstructured data, the imported data is analyzed using MapReduce and Pig to get the needed results and to answer on the research questions, the results were total number of crimes per years (2000-2014), state, type and gender (women). (Jain, A., & Bhatnagar, V., 2016) 0

3.3.2 NIFI Apache

NIFI is a dataflow system that can collect, transform, process and route data.it was built on flow-based programming concept, it was designed to automate and manage the flow of data between systems. (Peng, R., 2019)

NIFI is Java based and executed within JVM on a host operating system, as shown on figure 3 below the architecture of NIFI consist of different components, Web Server which is responsible about hosting NiFi's HTTP-based command and to enable the user to access NIFI via web based interface. Flow Controller which is responsible about providing and scheduling threads for execution. FlowFile Repository which is the area where NIFI track the status updates about the flowfiles. Content Repository that holds the content of flowfiles and Provenance Repository that holds provenance event data.

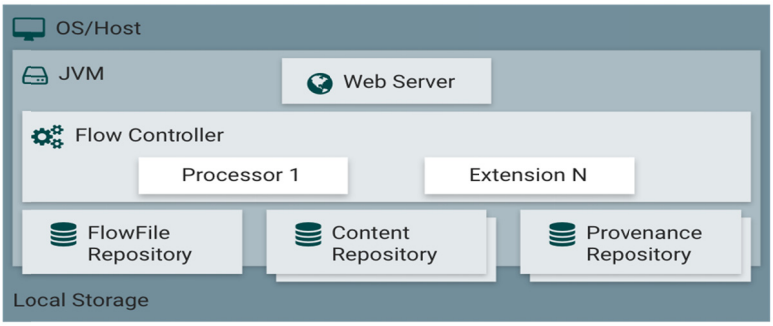

Figure 3. NIFI Architecture (Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S., 2018)

NIFI is able to run within a cluster, each node in the NIFI cluster complete the same tasks but interact with different set of data. The cluster is managed by cluster coordinator which is elected by ache Zookeeper.

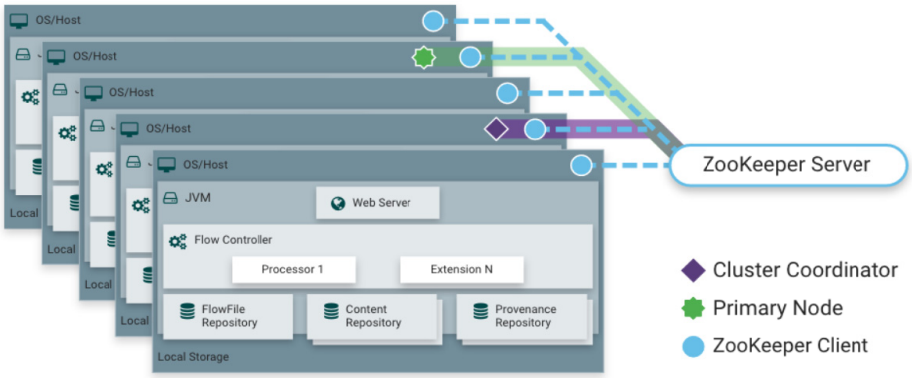

Figure 4. NiFi Distributed Architecture

NIFI has friendly web based user interface that allow users to drag and drop components to build the dataflow, the components can be started and stopped in real time also the errors and statistics can be viewed easily. NIFI buffering all the queued data and allow setting prioritization schemes to indicate how the data will be retrieved from the queue. NIFI provide data provenance module in order to track the data from the start of the flow until the end. The implemented dataflow is secure since NIFI use secured protocols like SSL, HTTPS, SSH and other encryptions.

A processor is an atomic element in NiFi dataflow which can do different tasks, it can read data from multiple resources, route, transform and publish data to external resources. For batch data ingestion NIFI processors can read the data from different sources, it can be any SQL database server like Oracle and MySQL, or NoSQL databases like MongoDB, or pulling data with different format from local or remote systems.

3.4 Stream Data Ingestion

Stream data or real time data is the data that comes out with the quick input and quick analysis to bring out a decision or action within a short time and very determined time line, the flow of data is very quick and difficult way that request to be managed, stored and analyzed there is a strong need to support real-time data ingestion particular for demanding new applications. Stream data ingestion is important for different sectors such as large volume of real time data in business needs to be ingested for developing mobile marketing analysis, advertising recommendation framework and visualizing the changed data and progress in real time. (Salah, H., Al-Omari, I., Alwidian, J., Al-Hamadin, R., & Tawalbeh, T., 2019)

Data Streaming Ingestion facing the challenges of processing the operational and real time data, which is vital in quick mutable situation. The process of streaming separates nonstop smooth input data into different units for advanced processing, when real time data is stored on hard discs will have a fair amount access of latency, so work with large volume of data makes hard discs are not suitable, which creates the memory challenge, existing systems often suffer from the extremely slow identification process.

Extra intensive data used to be extracted from all the data sources ranging from different live multimedia, to IoT data, and to real-time data from social media and blogs, growing applications of real time data analytics in the area of social media like Facebook and Twitter will create another challenge as the companies aim to ingest these data with low latency.

Security is another challenge for stream data ingestion process which comes out from quick growth of the internet, web-based systems who are facing malicious and suspicions files threatening in their security, so the ingestion process should provide security, auditing, and provenance. The analytical value from the stream data depends on accuracy and completeness of data so achieving good and accurate stream data ingestion is complicated and challenging task that require good planning and expertise (Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N., 217) (Pal, G., Li, G., & Atkinson, K., 2018) (Gurcan, F., & Berigel, M., 2018)

3.4.1 Flume Apache

It's a distributed reliable, available and efficient service for importing, collecting, aggregating and bringing in huge amount of data with its streaming feature and ingest it in a way that makes it easy for processing tool, hardly supports fault tolerance with accurate consistency ways, the data model used by flume is particularly used for online analytic application It has the most important role in data ingestion for real time data analytics, which is responsible for data refining and data visualization (Yadranjiaghdam, B., Pool, N., & Tabrizi, N., 2016) (Hemavathi, D., & Srimathi, H., 2017) (Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N., 2017) (Begum, N., & Shankara, A. A., 2016) Flume provides a framework for collecting and analyzing data from a sensor network with high performance scalability of HDFS. (Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N., 2017) (Ji, C., Liu, S., Yang, C., Wu, L., & Pan, L., 2016)

The data flow in flume same as pipeline that ingest data from the source to destination. Regarding to figure 5 below that discussed Flume architecture, data is transformed from source to destination based on flume agent which is JVM process that host the components during the data flow from the source to next end and it contains of channel, sink and the source.

Source is the part of agent that receive the data from related generators and move them to channel, the channel is considered as a bridge between the sink and sources, sink is the entity that sends the data to the destination. (Begum, N., & Shankara, A. A., 2016)

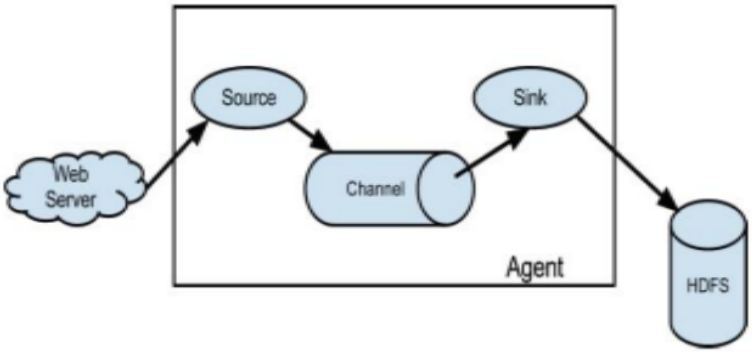

Figure 5. Flume Architecture

(Aravinth, S. S., Begam, A. H., Shanmugapriyaa, S., Sowmya, S., & Arun, E., 2015)

Flume was used in different researches, it was used in ingesting variance detection of household heating data "Jinan municipal steam heating system", sensors were built in all the rooms; this is to get information about the rooms like the thermal power, accumulated heat and temperature, within this research they depend on 16909 rooms of 394 buildings, flume is ingested all the sensors data, then all of these data will be processed by spark and come out with specified results (Lee, C., & Paik, I., 2017)

3.4.2 Kafka Apache

It's a distributed streaming tool that provides unified high real data feeds and messaging brokering system, the most important specification for Kafka is the low latency as all the process will occur in memory to prevent access latency of hard disks. (Shahin, D., Hannen Ennab, R. S., & Alwidian, J., 2019) Kafka provide high performance of ingestion large amount of messages with law latency and fault tolerance.

It has three major components broker, consumer and producer. Even if the consumer and producer were written in different programming language it will work efficiently and connecting different platforms together, not only used for streaming data other type of data also. Broker will be as the server in Kafka and it's responsible for the fault tolerance which is the most important feature in Kafka. Producer sends the message to consumer through broker, broker will be as a channel to differentiate the message. It has a high performance real time data channel (multi node and multi broker). (Lee, C., & Paik, I., 2017)

Based on the below figure 6, there are two main processes in Kafka Architecture first one distributing the messages and the second is publishing them. There are many servers in Kafka which processed as clusters, each one can deal with thousands of customers for huge amounts of read and writes capability as a central point for huge organization data, cluster in Kafka keeping the log of messages and give it a sequential ID. The ID is distributed through all the clusters and request a share of division which guarantee the fault tolerance (Hemavathi, D., & Srimathi, H., 2017) (Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N., 2017) (Pal, G., Li, G., & Atkinson, K., 2018) Kafka provides better scalability and message consistency compared to Flume. 0

The scalability feature in Kafka; allows the system to enlarge elastically and clearly without any stoppage. Data can be divided and distributed all over the machine even if the capacity of the machine is less than the size of data, the terminated messages are conserved on disk to avoid the data loss. (Hemavathi, D., & Srimathi, H., 2017) 00

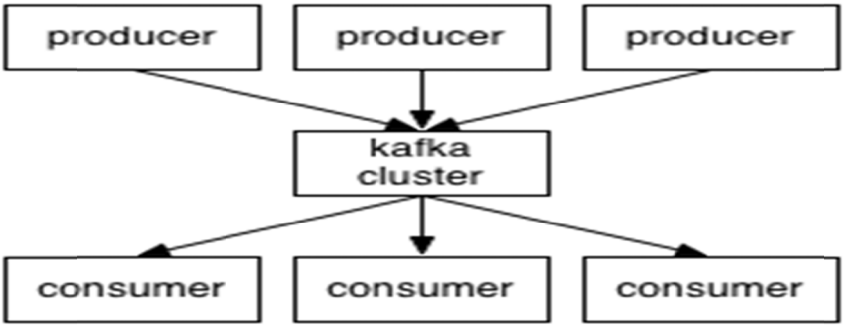

Figure 6. Kafka Architecture

Kafka was used in different researches, it was used to study the reaction of people about Tsunami earthquake in Japan, for this purpose they depend on Twitter social media, they depend on keywords like "Tsunami" and "Japan earthquake" during the tweets ingestion, the tweets are ingested, filtered, processed and visualized. Kafka was used for tweets ingestion and was linked to Twitter Streaming API, then the flow of tweets was classified based on their content. Spark was used for twitter data analysis and it was based on time, location of tweets and the time zone of tweeting. This information was processed in memory as its real time processing over massive amount of data flowing into the system, the results discuss how the people around the world reacted with Tsunami earthquake. (Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N., 2017)

3.4.3 NIFI Apache

NIFI is used for stream data ingestion also, it was built to solve the challenge of the flow data between systems. Flow means that some systems create the data and some of the systems consume the data.

NIFI is used in ingesting real time data; in many researches they depend on NIFI for ingesting the data into the system, one example that it was used media monitoring applications, like Twitter and safori channels. NIFI was used in a company with Kafka as a dataflow in a cluster. They used the live cloud based platform, which delivers messaging services labeled as RTM, the data is a real time data as input for the open data channels initiative. The first source of the data that need to be ingest from Big RSS and live data channels in Safori and as RSS feeds. This is as the biggest RSS aggregation in the world the volume of the data is over 6.5 million feeds3. The second source of the data and very vital source is from streaming news stories Twitter API platform, which offered a platform with scalable access to its data. They used two kinds of filtering tools as different number of filters as filtering capabilities for real time tweets and enterprise options with vital operator that has from 2502,000 filters with 2,048 character for each stream.

The big volume of the data and the high velocity for the data streams that come from Twitter streaming API depend of the reputation of the queries. The ingestion was using NIFI for all the flow with using three local processes groups. The filtered used to ingest media depends on removing the duplicates and noise data then the data will be routed to the related analytics systems. (Isah, H., & Zulkernine, F., 2018)

The below shown table compared all the ingestion tools that mentioned above which are Sqoop, Flume, NIFI and Kafka, based on different criteria's such as the latest stable release, type of loading, architecture and type of data that can be ingested and notable users for each tool.

All the mentioned tools are open source, written in Java and already used by notable users in the world, Sqoop is a good choice for ingesting batch data from RDBMS, it supports the data compression and has Bi-direction way from HDFS as it can import and export the data. NIFI is a recommended option for creating data flow between different systems, it can ingest both batch and stream data and provides back pressure, event prioritization and data compression properties. Kafka and Flume are dealing with streams data but Kafka is used for messaging purpose, Kafka is high scalable tool as different number of consumers can be added in an easy way.

Table 1. Comparison of Data Ingestion Tools

Criteria	Sqoop	Flume	NIFI	Kafka
latest stable release	1.4.7	1.9.0	1.10.0	2.4.0
Primary written in	Java	Java	Java	Java
License	Open Source	Open Source	Open Source	Open Source
Basic nature	works well with any RDBMS that has JDBC(Java Database connectivity) like oracle	works well for streaming data sources which continuously generating.	works well for data flow creation between different systems.	works well for messaging Streaming data
Type of Data	Batch Data	stream Data	Batch and Stream Data	Stream
Type of loading	Not-event driven	Event driven	Both (Event and not-event)	Event driven
Architecture	Connector based	Agent based	Flow based	Process topology .
Link to HDFS	Connected	Connected	Connected	Programmable
Direction based on HDFS	Bi-directional	Uni-Directional (into Hadoop)	Bi-directional	-
User Interface	Shell command line	Shell command line	Web user interface	Shell command line
Data compression	Supported	Supported	Supported	Supported
Event prioritization	-	Supported by (Failover sink processor) concept	Supported	Programmable
Back pressure	-	No	Yes	Yes
Notable users	1-Apollo Group 2-Coupons.com	1-Meebo . 2-Sharethrough . 3-SimpleGeo.	1-Macquarie Telecom 2-Group. 3-Hastings Group. 4-Payoff.	1-Uber 2-Booking.com 3-Slack.

4. Data Preparation

For Big Data Analytics as shown in figure 7 below, the data preparation stage is considered as the most integral phase in which data preprocessing and integration operations are performed in order to enhance big data quality and suitability. This phase embraces a wide range of operations and techniques that are mainly applied to generate useful data sets for further data mining algorithms.

For example, in the real-world, the collection of big data from various sources such as sensors and social media using the Internet-of-Things (IoT) techniques will produce massive data with irrelevant and noisy information. Therefore, tackling the problem of noisy data, outliers, and anomalies are required to provide noise-free and high-quality datasets.

Furthermore, at this level of data cleansing and de-noising, it is imperative to deploy feature extraction methods to separate useful and structured data from big data rows. Other challenges in the data preparation phase will appear depending on the nature of big data sources including the velocity and variety of big data types. As the gathered data from various sources will differ in data type, format and dimension accordingly, intelligent data fusion process, dimension reduction and uniform datasets techniques are performed for achieving data integrity and consistency for the collected unstructured and semi-structured data streams.

However, the presence of missing data values that cannot be avoided in data analysis remains a huge issue even with the creation of a uniformly structured big data format. Thus, it is necessary to deploy operations that handle missing values such as data elimination, sketching, and imputation to increase the efficiency of knowledge extraction processes and improve the overall quality of produced data for better decision making. (Rehman, M. H. ur, Chang, V., Batool, A., & Wah, T. Y., 2016)

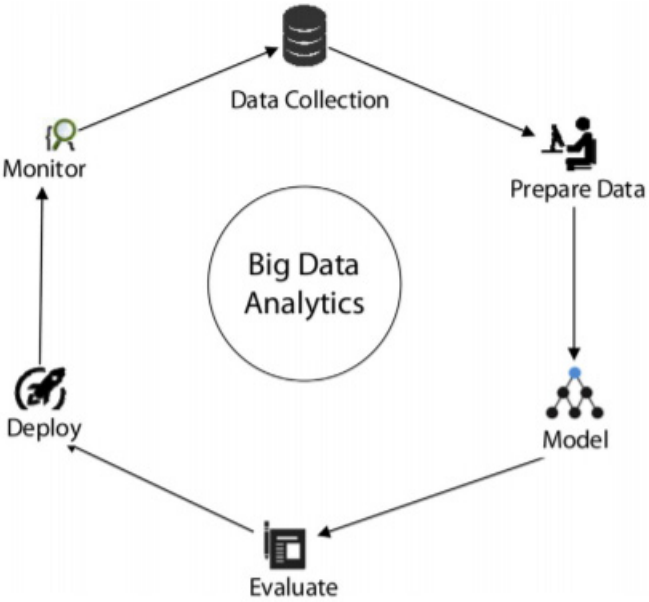

Figure 7. Big Data Analytics (Rallapalli, S., & Gondkar, R. R., 2015)

4.1 Spark Apache

Apache Spark is a framework help to improve the speed of processing by the distribution technique and using the memory instead of disk. This platform allows user programs to transfer data into memory and repeated the queries; this feature makes it a better choice tool for online and reiterated processing (such as ML algorithms). It inspired by the limitations in the MapReduce/Hadoop apache. (Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., … Zaharia, M., 2015) (Abuqabita, F., Al-Omoush, R., & Alwidian, J., 2019)

Spark is based on Resilient Distributed Datasets (RDDs) which are unalterable and divided groups of records, have a programming interface for acting transformations operations (such as map, filter and join, union) on over multiple data items, or actions operations (such as reduce, count, first, and so on) that return a value after operate a computation on an RDD. Spark saves all transformations that execute to build a dataset for fault-tolerance purposes.

The spark architecture as discussed in the below figure divided into four main parts first of all Spark SQL which is given a possibility to queries data through a command-line interface and ODBC/JDBC controllers. Spark streaming which allows us to use Spark's API in streaming environments by divided the batches to small ones which means quick processing, it can work with several data sources like HDFS, Flume or Kafka. Machine learning library (MLlib) has an important task like classification, regression, clustering, optimization, and dimensionality reduction, this library has been designed to simplify ML pipelines in large-scale environments. Finally, we have Spark GraphX which is the graph processing system in Spark.

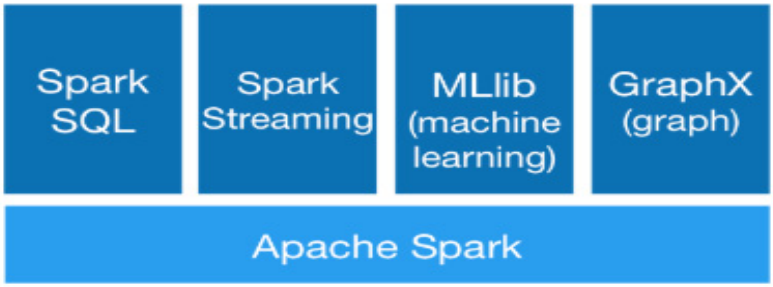

Figure 8. Spark Architecture

Many features that characteristics spark, the important one is speed to run an application in Hadoop cluster, more than 100 times faster when it runs in memory, and 10 times faster when it runs on disk. This is achieved by reducing the number of read and write operations on disk, it stores the temporary processing data in memory. Spark supports multiple languages such as Java, Scala, or Python.

Spark was used into a study in 2019 goals to analysis an agricultural big data, plenty of terminal equipment in the agricultural park collect environmental data that affects crop growth every day.

Hadoop and Spark are used for improving this analysis. They developed applications for real agricultural park big data analysis in both frameworks and implemented a yield prediction model based on multiple linear regression using Spark MLlib. The results show that the performance of Spark is higher than Hadoop, and the model can obtain better prediction results. 0(Cheng, Y., Zhang, Q., & Ye, Z., 2019)

4.2 Hive Apache

Hive is a Data warehouse system for Hadoop. It similar to any SQL language it runs queries that compiled and deal with functions as MapReduce and return the result to the user. Hadoop has unstructured data that has some unclear structure connected with it, the important reason for using Hive that easy to work on the Hadoop file system and MapReduce for non-developers. Users like scientists, analysts. Who already know SQL syntax, they can find out the data by writing SQL statements instead of writing code which means less time. (Surekha, D., Swamy, G., & Venkatramaphanikumar, S., 2016)

Hive perfect when its use with data aggregation method, Adhoc querying, and analysis a massive data such as Analyzing social media data such as Twitter Data (Bhardwaj, A., Vanraj, Kumar, A., Narayan, Y., & Kumar, P., 2015).The below diagram is the architecture of the hive. Hive metadata (or called metastore) can use embedded, local, or remote databases. Hive servers are built on Apache economy Server technology. A multi of user interfaces and web UI give another advantage of using Hive.

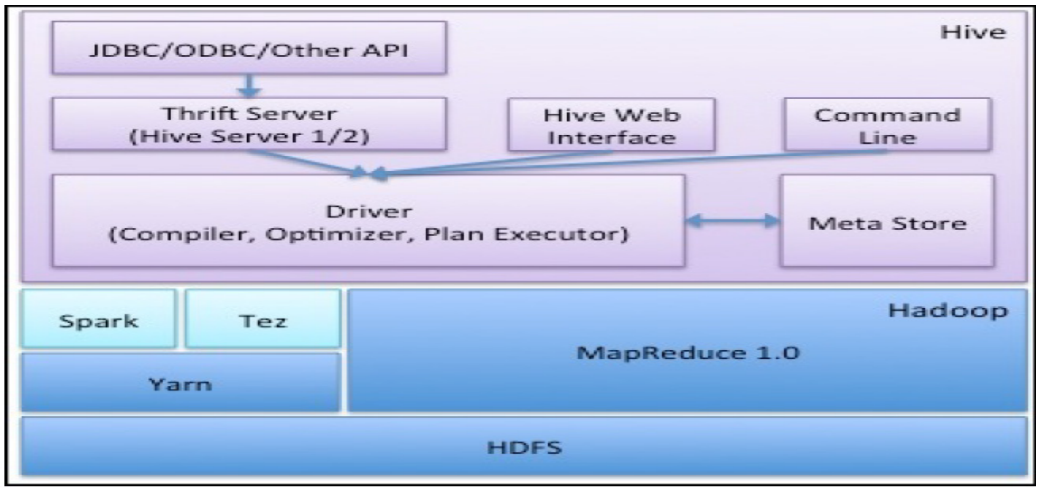

Figure 9. Hive Architecture

Hive was built on top of Hadoop, it simplifies access to data via SQL, thus enabling data warehousing tasks such as extract/transform/load (ETL), reporting, and data analysis. Easy Access to files stored either directly in HDFS or in other data storage systems such as HBase.

In August 2016 hive was applied as a new big data tool because challenges of the traditional database systems to face the differences in nature and complexity with the data obtained from multi sources. They performance profiling of Meteorological and Oceanographic data on Hive is conducted. Hive being the commonly used data warehouse analytical platform for big data is chosen with the view to exposing the intricacies that are involved in the formatting and loading of the data. The Meteorological and Oceanographic data if properly formatted its analytics with Hive proved to be efficient compared to the traditional database systems. The results of this study have the potentials of attracting the oil and gas companies to adopt big data technologies for the handling of their exploration dataset. (Abdullahi, A. U., Ahmad, R., & Zakaria, N. M., 2016)

4.3 Impala Apache

Impala is a SQL query tool on big data designed for real-time processing its interactive and responsive tools, which is inspired by the Google Dremel project and developed by Cloudera. Impala using different approaches to deal with data does not like Hive and MapReduce, it uses a distributed query engine instead of slow batch processing mode, and the distributed engine is similar to the parallel relational database. It can use all the features on SQL and other statistical functions to apply it to the data stored in HDFS or HBase directly, this represent another value on reducing the delay. (Jingmin Li, 2014)

Regarding to the below figure, Impala consists of two main parts, the first one called Impalad which is a distributed query process and it consists of query planner, query coordinator, and query execution engine. The second part is Impala state which a process is called statestored which is responsible for collecting CPU/memory/network resource information from all nodes in the cluster. It creates multiple threads to process the Impalad subscription and keep the heartbeat connection with each Impalad. The Impalad will cache a copy of the information in the statestored and swipe its mode between recovery mode and normal mode depends on statestored status (offline/online). (Jingmin Li, 2014)

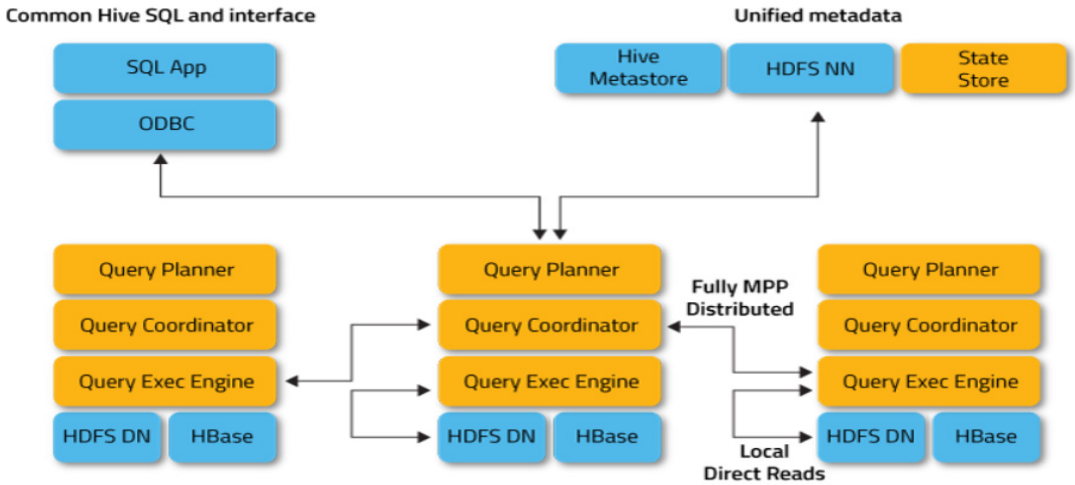

Figure 10. Impala Architecture

Impala supports important Features such as familiar SQL interface that data scientists and analysts can query a massive amount of data on Hadoop, single system for big data processing and analytics, so users can have less costly choices modeling and ETL for analytics, also simplicity on dealing with other Apaches like sharing data files between different components with no copy or export/import step.

Impala used in different researches, it was used in Wireless Sensor Network (WSN) in 2016 which is a system that has a capability to conduct data acquisition and monitoring in a wide sampling area for a long time and its need a big data because the massive data that generated from WSN so the traditional database system cannot be able to handle this. Big data provide a high data storage system and data analysis process. They developed a WSN system for CO2 monitoring using Kafka and Impala to distribute a huge amount of data. Sensor nodes

gather data and accumulated in temporary storage then streamed via Kafka platform, and these data stored using Impala database. (Wiska, R., Habibie, N., Wibisono, A., Nugroho, W. S., & Mursanto, P., 2016)

4.4 Storm Apache

Storm defines as a distributed system for computing and real-time data processing it's a fault-tolerance and easy compile complex real-time computation in a computer cluster, it's similar to what Hadoop does in batch processing. Storm can ensure that the messages will be speedily systematic. Additionally, it integrated with many program languages for development. (Yang, W., Liu, X., Zhang, L., & Yang, L. T., 2013)

Regarding to the below figure, storm cluster divided into three nodes: Nimbus node (master node) which help to uploads computations for execution, distributes code across the cluster, launches workers across the cluster, supervises computation and reallocates workers as needed, the second is Zookeeper nodes – coordinates the Storm cluster, at the end, we have supervisor nodes – communicates with Nimbus through Zookeeper, starts and stops workers according to signals from Nimbus.

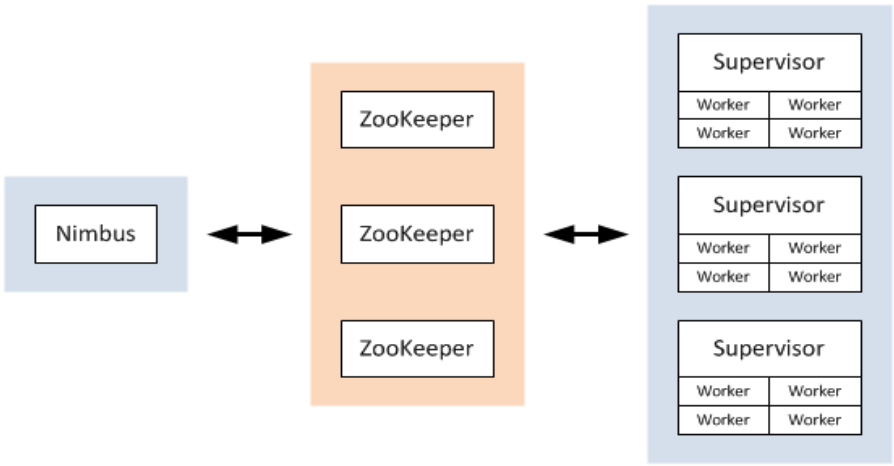

Figure 11. storm cluster

Storm process data using different components, tuples for an ordered list of elements, streams for an unbounded sequence of tuples, spouts to define sources of streams in a computation, bolts process input streams and produce output streams. They can run functions such as filter, aggregate, or join data, or contact to databases to create topologies that give the overall calculation, represented visually as a network of spouts and bolts.

The main Advantages of Storm that is a simple programming model with High fault-tolerance and parallel computation, and it have a service framework support hot deployment, reliable message handling and finally has "Local Mode" simulation. (Yang, W., Liu, X., Zhang, L., & Yang, L. T., 2013)

Storm used in 2015 on abstract real-time GIS data model and sensor web service platform, it was proposed to manage real-time environmental data. With the development of sensor technology, and regarding to the large accumulative sensor networks environmental data. The study integrates Strom with the sensor web service and processing the environmental data timely. To test the feasibility of the design and implementation, two use cases of real-time air quality monitoring and real-time soil moisture monitoring based on the real-time GIS data model in the sensor web service platform are realized and demonstrated. The experimental results show that the implementation of real-time GIS data model and sensor web service platform with the Apache Storm is an effective way to manage real-time environmental big data. 0

The below shown table is a comparison for data preparation tools that mentioned above which are Hive, Impala, Spark and Storm. The comparison is based on different criteria like the latest stable release, scope, license and use case, all the mentioned technologies are open source but developed by different companies, Hive and Impala are SQL based tools, storm is stream based and Spark is for SQL, stream, Graph and ML.

Both Hive and Impala is SQL engines but Impala is faster than Hive, Impala is a good choice for BI analytics queries on Hadoop as it provides low latency and high concurrency, however Hive is used for building an efficient data warehousing solutions.

Spark is a fast processing engine that provides different capabilities such as interactive analytics, streaming data, and machine learning so it is a good choice for online and real-world applications.

Table 2. Comparison of Data Preparation Tools

Criteria	Hive	Impala	Spark	Storm
Latest stable release	3.1.2	2.2.0	2.4.4	0.2.37
Main Backers	Facebook	Cloudera	AMPLab	BackType
License	Open Source	Open Source	Open Source	Open Source
Scope	SQL (HiveQL)	SQL	Stream, Batch, Graph, ML, SQL	Stream
Primary database model	Relational RDBMS	Relational RDBMS	Relational RDBMS	-
SQL File format	1-Sequence File 2-RCFile 3-Avro Files 4-ORC Files 5-Parquet	1-Parquet 2-Text File 3-(Avro ,RC File and Sequence file only for table creation)	1-Parquet Files 2-ORC Files 3-JSON Files 4-Avro Files	-
Table Partitioning	Supported	Supported	Supported	-
Querying Latency	High	Low	Low	-
Stream Processing	-	-	Supported	Supported (micro-batch processing)
Stream Primitives	-	-	DStream	Partition, Tuples
Stream Sources	-	-	HDFS, DBMS, Kafka and other sources	Spout
Streaming query	-	-	Yes (spark SQL)	No
Use Case	Data warehousing	BI Style queries.	Streaming Data Machine Learning Interactive Analysis	Streaming Data

5. Conclusion and Future Work

As mentioned above the format of data is varied from different sources and the volume of generated data become very large. To start any Big Data project, the data should be ingested and prepared for the processing. This paper presented a review about Big Data ingestion and preparation concepts and mentioned some tools for each process, each tool was discussed with its characteristics and real use case, a comparison between the data

ingestion tools was discussed in table 1 in order to help users to choose the tool that satisfies their needs, also comparison between data preparation tools was reviewed under table 2.

In future research, the performance indicator such as speed and number of processed files per minute will be studied for each tool and mentioned within the comparison.

References

Abdullahi, A. U., Ahmad, R., & Zakaria, N. M. (2016). Big data: Performance profiling of Meteorological and Oceanographic data on Hive. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS). https://doi.org/10.1109/ICCOINS.2016.7783215

Abuqabita, F., Al-Omoush, R., & Alwidian, J. (2019). A Comparative Study on Big Data Analytics Frameworks, Data Resources and Challenges. *Modern Applied Science, 13*(7), 1. https://doi.org/10.5539/mas.v13n7p1

APACHE HIVE. Retrieved December 2019, from https://hive.apache.org/

Apache Sqoop. Retrieved December 2019, from https://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html

Aravinth, S. S., Begam, A. H., Shanmugapriyaa, S., Sowmya, S., & Arun, E. (2015). An efficient HADOOP frameworks SQOOP and ambari for big data processing. *International Journal for Innovative Research in Science and Technology*. ISSN (online): 2349-6010

Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., … Zaharia, M. (2015). Scaling spark in the real world. *Proceedings of the VLDB Endowment, 8*(12), 1840–1843. https://doi.org/10.14778/2824032.2824080

Begum, N., & Shankara, A. A. (2016). Rectify and envision the server log data using apache flume. *Int. J. Technol. Res. Eng, 3*(9). ISSN (Online): 2347 - 4718

Bhardwaj, A., Vanraj, Kumar, A., Narayan, Y., & Kumar, P. (2015). Big data emerging technologies: A CaseStudy with analyzing twitter data using apache hive. 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS). https://doi.org/10.1109/RAECS.2015.7453400

Bucur, C. (2015, July). Using big data for intelligent businesses. In Proceedings of the Scientific Conference AFASES (Vol. 2, pp. 605-612).

Chen, Z., Chen, N., & Gong, J. (2015). Design and implementation of the real-time GIS data model and Sensor Web service platform for environmental big data management with the Apache Storm. 2015 Fourth International Conference on Agro-Geoinformatics (Agro-Geoinformatics). https://doi.org/10.1109/Agro-Geoinformatics.2015.7248139

Cheng, Y., Zhang, Q., & Ye, Z. (2019). Research on the Application of Agricultural Big Data Processing with Hadoop and Spark. 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). https://doi.org/10.1109/ICAICA.2019.8873519

Erraissi, A., Belangour, A., & Tragha, A. (2018). Meta-Modeling of Data Sources and Ingestion Big Data Layers. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3185342

Gurcan, F., & Berigel, M. (2018). Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges. 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). https://doi.org/10.1109/ISMSIT.2018.8567061

Hadoop, A., Cloudera, Inc, & Apache Software Foundation. (2020, January 14). Cloudera. Retrieved December 2019, from https://www.cloudera.com/

Hemavathi, D., & Srimathi, H. (2017). Survey on data failure handling methods of streaming data. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS). https://doi.org/10.1109/ICCONS.2017.8250686

Isah, H., & Zulkernine, F. (2018). A Scalable and Robust Framework for Data Stream Ingestion. 2018 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/BigData.2018.8622360

Jain, A., & Bhatnagar, V. (2016). Crime Data Analysis Using Pig with Hadoop. *Procedia Computer Science, 78*, 571–578. https://doi.org/10.1016/j.procs.2016.02.104

Ji, C., Liu, S., Yang, C., Wu, L., & Pan, L. (2015). IBDP: An Industrial Big Data Ingestion and Analysis Platform and Case Studies. 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI). https://doi.org/10.1109/IIKI.2015.55

Jingmin Li. (2014). Design of real-time data analysis system based on Impala. 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA). https://doi.org/10.1109/WARTIA.2014.6976427

Lee, C., & Paik, I. (2017). Stock market analysis from Twitter and news based on streaming big data infrastructure. 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST). https://doi.org/10.1109/ICAwST.2017.8256469

MATACUTA, A., & POPA, C. (2018). Big Data Analytics: Analysis of Features and Performance of Big Data Ingestion Tools. Informatica Economica, 22(2/2018), 25–34. https://doi.org/10.12948/issn14531305/22.2.2018.03

Meehan, J., Aslantas, C., Zdonik, S., Tatbul, N., & Du, J. (2017, January). Data Ingestion for the Connected World. In CIDR.

Mohamed, E., & Hong, Z. (2016). Hadoop-MapReduce Job Scheduling Algorithms Survey. 2016 7th International Conference on Cloud Computing and Big Data (CCBD). https://doi.org/10.1109/CCBD.2016.054

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., & Arocena, P. C. (2019). Data lake management. *Proceedings of the VLDB Endowment, 12*(12), 1986–1989. https://doi.org/10.14778/3352063.3352116

Oughdir, L., Dakkak, A., & Dahdouh, K. (2019). Big data: a distributed storage and processing for online learning systems. *International Journal of Computational Intelligence Studies, 8*(3), 192. https://doi.org/10.1504/IJCISTUDIES.2019.10024283

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences, 30*(4), 431-448. https://doi.org/10.1016/j.jksuci.2017.06.001

Pal, G., Li, G., & Atkinson, K. (2018). Big Data Ingestion and Lifelong Learning Architecture. 2018 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/BigData.2018.8621859

Peng, R. (2019). Kylo Data Lakes Configuration deployed in Public Cloud environments in Single Node Mode. DiVA, id: diva2:1367021

Rallapalli, S., & Gondkar, R. R. (2015). Map reduce programming for electronic medical records data analysis on cloud using apache hadoop, hive and sqoop. *International Journal of Latest Technology in Engineering, Management & Applied Science*. ISSN 2278.

Rehman, M. H. ur, Chang, V., Batool, A., & Wah, T. Y. (2016). Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management, 36*(6), 917–928. https://doi.org/10.1016/j.ijinfomgt.2016.05.013

Salah, H., Al-Omari, I., Alwidian, J., Al-Hamadin, R., & Tawalbeh, T. (2019). Data Streams Curation for Better Machine Learning Functionality and Result to Serve IoT and other Applications: A Survey. *Journal of Computer Science, 15*(10), 1572–1584. https://doi.org/10.3844/jcssp.2019.1572.1584

Shahin, D., Hannen Ennab, R. S., & Alwidian, J. (2019). Big Data Platform Privacy and Security, A Review. IJCSNS, 19(5), 24.

Surekha, D., Swamy, G., & Venkatramaphanikumar, S. (2016). Real time streaming data storage and processing using storm and analytics with Hive. 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT). https://doi.org/10.1109/ICACCCT.2016.7831712

Team, A. N. F. (n.d.). Apache NIFI. Retrieved December 2019, from https://nifi.apache.org/docs/nifi-docs/html/user-guide.html

Wiska, R., Habibie, N., Wibisono, A., Nugroho, W. S., & Mursanto, P. (2016). Big sensor-generated data streaming using Kafka and Impala for data storage in Wireless Sensor Network for CO<inf>2</inf> monitoring. 2016 International Workshop on Big Data and Information Security (IWBIS). https://doi.org/10.1109/IWBIS.2016.7872896

Yadranjiaghdam, B., Pool, N., & Tabrizi, N. (2016). A Survey on Real-Time Big Data Analytics: Applications and Tools. 2016 International Conference on Computational Science and Computational Intelligence (CSCI). https://doi.org/10.1109/CSCI.2016.0083

Yadranjiaghdam, B., Yasrobi, S., & Tabrizi, N. (2017). Developing a Real-Time Data Analytics Framework for Twitter Streaming Data. 2017 IEEE International Congress on Big Data (BigData Congress). https://doi.org/10.1109/BigDataCongress.2017.49

Yang, W., Liu, X., Zhang, L., & Yang, L. T. (2013). Big Data Real-Time Processing Based on Storm. 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. https://doi.org/10.1109/TrustCom.2013.247