International Journal of Engineering & Technology, 7 (2.7) (2018) 577-588

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Parallel processing on Big Data in the context of Machine Learning and Hadoop Ecosystem: A Survey

Anilkumar Vishwanath Brahmane[1]*, R. Murugan[2]

[1]Department of Computer Science and Engineering, K L E F Deemed to be University, A.P., India
[2]Department of Computer Science and Engineering, K L E F Deemed to be University, A.P., India
**Corresponding author E-mail: vb_anil@yahoo.co.in*

Abstract

Emergent Big Data applications have become gradually more essential. In reality, a lot of institutes, businesses and in general entire society from diverse segments depend more and more on information take out from enormous quantity of raw information, statistics and numbers. On the other hand, in Big Data perspective, customary information methods and policies are not as much of capable. They prove a time-consuming receptiveness and are short of quantifiability, measurability, presentation and accurateness. To solve the composite Big Data constraints and difficulties, a large amount effort has been carried out. As an effect, different categories of packages, distributions and technologies have been developed. In this paper an evaluation is done, this studies recent technologies developed for Big Data. It aims to assist to choose and adopt the exact combination of diverse Big Data technologies according to their technological, scientific needs and particular applications requirements. It provides not only a worldwide sight of most important Big Data technologies but also relationship according to special organizational, classifications levels such as Information Storage Level, Information Processing Level, Information Querying Level, Information Access Level and Management Level. It classifies and talks about main tools and its features, advantages, restrictions and treatments.

Keywords: *Big Data; Hadoop; Machine Learning; Parallel Processing.*

1. Introduction

Currently, huge data volumes are every day generated at extraordinary speed from various foundations. This is because of numerous technological and scientific developments, together with the IoT, the explosion of the Cloud Computing [1] as well as the increase of smart devices. At the back scene, dominant systems and distributed applications are behind such multiple connections systems e.g., smart grid systems, healthcare systems, retailing systems like that of Walmart, government systems etc.

Earlier to Big Data upheaval, industries could not accumulate all their records and documentation for long periods. They could not powerfully deal with immense informational index.

Undoubtedly, customary tools have restricted storage space, rigid managing tools and are costly. Versatility, adaptability and execution are very much essential factors in Big Data context. Resources, methods and powerful technologies plays an significant role in Big Data management. Big Data require to clean, process, analyze, secure and provide a granular access to monstrous advancing informational collections.. Corporations and industry are alert that data investigation is becoming a very important issue.

Different countries have significant schemes. In March 2012, the USA government initiates Big Data Research and Development Initiative [2]. In Japan, Big Data development became one important axe of the national technological strategy in July 2012 [3]. The United Nations released a information Big Data for Development: Opportunities and Challenges [4]. Because of this, diverse Big Data projects, models, frameworks and innovative technologies were produced to offer extra storage space, parallel processing and problem solving analysis of heterogeneous systems. Many systems are developed for data privacy, protection and legitimacy These systems and solution are very good in term of flexibility, scalability and performance. And the cost of hardware and processing solutions is always reducing because of new technological advancement [5].

More importantly knowledge, facts, and information are essential factors one needs to extract from Big Data. As well more accurate results for Big Data applications are also important. So many advance models are proposed. It's very challenging task to select the appropriate models. One cannot ignore the various aspects like security, cost, reliability, efficiency, technical compatibility, performance supports and deployment complexity. After all this factors, algorithms and techniques are the most dominant factors used to process Big than technologies. In this paper, we present a survey on recent technologies developed for Big Data. We categorize and deeply compare them according to their usage, benefits, limits and features. While categorizing and classifying these Big Data technologies different phases are consider like Information Storage Layer, Information Processing Layer, Information Querying Layer, Information Access Layer and Management Layer. This helps to better understand the connections among various Big Data technologies and their functioning.

This paper is organized as follows. Section 2 defines Big Data and presents some of its applications. Section 3 identifies and discusses some technical challenges faced in dynamic Big Data environment. Section 4 presents Hadoop framework and compares some main modules developed on top of it (e.g., data storage, integration, processing and interactive querying). Section 5 presents main Hadoop distributions.

2. Background

2.1. Big Data definitions

Unlike traditional data, the term Big Data is large data containing structured, unstructured and semi-structured data. Big Data has a complex nature that requires powerful technologies and advanced algorithms. So the traditional static Business Intelligence tools can no longer be efficient in view of applications related to Big Data.

Most data scientists and experts define Big Data by the following seven main characteristics (called the 7Vs)

Volume: Immense quantity of digital information is produced continuously from millions of devices and applications (ICTs, smart-phones, products codes, social networks, sensors, logs, etc.). Research studies estimated that about 2.5 Exabyte were generated each day in 2012. This amount is doubling every 40 months approximately. 4.4 ZB digital data as per records given by International Data Corporation, in 2013, were produced, simulated, consumed, and replicated. This is exactly doubles every 2 years. This figure reached to 8 ZB in 2015. In future this figure will reach to 40 Zeta bytes.

Velocity: The speed of data generation is very high. For instance, Wallmart transactions can givens more than 2.5 PB of data per hour. YouTube, Face book are another examples which produce data with very high speed.

Variety: Medium, sources, formats and context are different for different types of Big Data.

Vision, **Verification, Validation**, **Value** are other Vs in Big Data context.

2.2. Big Data applications

Here are some examples of Big Data applications.

Smart Grid case: For national electronic power consumption, Smart grids operations can plays important roles. Many connections among smart meters, sensors, control centers and other infrastructures produce huge amount data. With the help of Big Data analytics one can identify at-risk transformers and to detect abnormal behaviors of the connected devices. Energy-forecasting analytics help to better manage power demand load, to plan resources, and hence to maximize prots [61].

E-health: Big Data is generated from different heterogeneous sources likes' laboratory and clinical data, patients symptoms uploaded from distant sensors, hospitals operations, and pharmaceutical data. This helps to health services. Public health plans as per population symptoms, disease evolution. To optimize hospital functioning and to decline health cost [61].

Internet of Things (IoT): The biggest market of Big Data applications is captured by IoT. To track vehicles positions with sensors, wireless adapters, and GPS. This information can be used to supervise and manage employees. To minimize delivery routes. Smart city is one of the good and challenging research based on the application of IoT data [61].

Public utilities: In Water supply department to identify leakages, illegitimate water connections and remotely manage valves to make sure fair supply of water to different regions of the city [61].

Transportation and logistics: RFID and GPS system can track the vehicle. Information related to For instance, data collected about the number of passengers using the buses in different routes are used to optimize bus routes and the frequency of trips. Passenger's recommendations with valuable information to find next bus to find shortest path towards destination. Mining Big Data helps also to get better travelling company by forecasting order about public or private networks [61].

Political services and government monitoring: To supervise political movements and analyse population emotions or feelings. Social networking, interviews, and voting are various means to get political and government related data. National and local problems can be identified by such systems [61].

3. Big Data challenges

The mining of Big Data offers many attractive opportunities. However, researchers and professionals are facing several challenges when discovering Big Data and when taking out value and knowledge from such mines of information. The difficulties lye at different levels [61] including: information fetch, storage, penetrating, distribution, investigation, organization and revelation. Furthermore, there are security and privacy issues especially in distributed data driven applications. Often, the deluge of information and distributed streams surpass our capability to harness. The size of Big Data is growing exponentially. Present models can only handle the Big Data in PB, ZB or EB. In this section, we discuss in more details some technological issues still opened for research.

3.1. Big Data management

Data scientists are facing many challenges when dealing with Big Data. The main fact is how to gather, put together and accumulate this tremendous data with limited hardware and software requirements [3] [6]. Another challenge is Big Data management. It is crucial to efficiently manage Big Data in view to facilitate the extraction of reliable insight as well as to optimize expenses. Indeed, a good data management is the foundation for Big Data analytics. Big Data management means to clean data for reliability, to aggregate data coming from different sources and to encode data for security and privacy. It means also to ensure efficient Big Data storage and a role-based access to multiple distributed endpoints. In other words, Big Data management goal is to ensure reliable data that is easily accessible, manageable, properly stored and secured.

3.2. Data aggregation

One more challenge is to coordinate outside data sources and distributed Big Data with the internal infrastructures of an organization. Most of the time, it is not sufficient to analyze the data generated inside organizations. In order to extract valuable insight and knowledge, it is important to go a step further and to aggregate internal information with external information sources.

3.3. Imbalanced system capabilities

An important issue is related to the computer architecture and capacity. If you consider the Moore's Law about processor design, the I/O operations may hamper because of mismatch in the performance pattern [8]. Consequently, this imbalanced system capacity may slow accessing data and affects the performance and the scalability of Big Data applications. From another angle, we can notice the various devices capacities over a network (i.e., sensors, disks, memories). This may slow down system performance.

3.4. Imbalanced Big Data

Another challenge is classifying imbalanced dataset. In fact, real-world applications may produce classes with different distributions. The first type of class those are under presented with insignificant amount of occurrences (known as the minority or positive class). The second class that have an rich amount of occurrences. Identifying the minority classes is important in various fields such as medicinal analysis [9], software faults detection [10], Finances [11], drug discovery [11] or bio-informatics [12].

The classical learning techniques are not adapted to imbalanced data sets. This is because the model construction is based on global search measures without considering the number of instances. Indeed, global rules are usually privileged instead of specific rule so the many of the class are abandoned throughout the model building. Thus, Standard learning techniques do not consider the dissimilarity among the amount of samples fit in to dissimilar classes [13]. However, the classes which are under-represented may constitute important cases to identify.

Protein fold classification and weld flaw classification [14] having more than two classes. These create new test that are not experimental in two-class problems. Two categories are useful in solving such problems. Binary classification e.g., discriminant analysis, decision trees, k-nearest neighbors, Naive Bayes, neural networks, and support vector machines. Decomposition and Ensemble Methods (DEM). This can make the use of Binary Classifiers (BCs), and then obtaining a new observation with the help of BCs predictions [15].

3.5. Big Data analytics

Superior information analysis is essential to recognize the associations among features and explore data. Therefore, Superior algorithms and competent techniques of data mining are needed to get exact outcomes, to check the changes in different fields and to forecast upcoming remarks. Though, big data analysis is at rest not easy for many reasons: the composite nature of Big Data together with the 5Vs, the need for scalability and performance to examine such incredible mixed data sets with real-time sensitivity [16] [17]. Today, a variety of analytical techniques together with data mining, visualization, statistical-numerical-arithmetical analysis, and machine learning. A lot of studies deal with this region by enhancing the used techniques, proposing new ones or testing the combination of various algorithms and technologies. As a result, Big Data pressed the growth of systems architectures, the hardware as well as softwares. But, we at rest require analytical advancement to focus Big Data challenges and stream processing. How to promise the timeliness of reply while the amount of data is very large?

4. Big Data Machine Learning

The purpose of machine learning is to find out knowledge and make smart and sharp decisions. Examples are [18];

1. Recommendation engines,
2. Recognition systems,
3. Informatics and data mining, and
4. Autonomous control systems.

Generally, the Field of Machine Learning (ML) is divided into;

1. Supervised learning,
2. Unsupervised learning,
3. Reinforcement learning.

4.1. Data Stream learning

Current real-world applications such as sensors networks, credit card transactions, stock management, blog posts and net-work traffic produce tremendous datasets. Data mining methods are important to discover interesting patterns and to extract value hidden in such huge datasets and streams. Table 1 shows the data mining algorithms.

Table 1: Traditional Data mining algorithms

Algorithms	Challenges
Association mining	Lack of efficiency , scalability & accuracy when applied to Big Data
Clustering	
Classifications	

Because of the size, speed and variability of streams, it is not feasible to store them permanently then to analyze them. Thus researchers need to find new ways to optimize analytical techniques, to process data instances in very limited amount of time with limited resources (i.e., memory) and to produce in real-time accurate results.

Furthermore, variability of streams brings unpredictable changes (i.e., changing distribution of instances) in incoming data streams. This concept drift affects the accuracy of classification model trained from past instances. Therefore, several data mining methods were adapted to include drift detection techniques and to cope with changing environment. Classification and clustering are the most studied ones.

Experiments on data streams demonstrated that changes in underlying concept affect the performance of classifier model. Thus, improved analytical methods are needed to detect and adapt to the concept drifts [19].

As an example in the current unstable economic environment, enterprises need an efficient Financial Distress Predict (FDP) system. Such system is crucial to improve risk management and support banks in credit decisions. DFDP (Dynamic Financial Distress Prediction) became an important branch of FDP research [20]. It improves corporate financial risk management. It focuses on how to update the FDP model dynamically when the new sample data batches gradually emerge and FDC (Financial Distress Concept drift) happens over time.

4.2. Deep learning

Deep learning is a active research field in machine learning and pattern recognition. Important role in;

1. Computer vision
2. Speech recognition
3. Natural language processing [3].

Conventional machine-learning techniques and feature engineering algorithms, are having limitations to process natural data [21]. Deep Learning is more commanding to solve data analytical and learning problems. To automatically extracting complex data representations from large volumes of unsupervised and uncategorized raw data.

This is the hierarchical learning and extraction of several different layers composite data. This is suitable to simplify the analysis of;

1. Large data volumes,
2. Semantic indexing,
3. Data tagging,
4. Information retrieval,
5. Discriminative tasks

Big Data at rest faces considerable tests to deep learning [6]:

1. Huge volumes of Big Data
2. Heterogeneity
3. Noisy labels, and non-stationary distribution.
4. High velocity:

Big scope for

- How to improve Deep Learning algorithms in order to tackle

1. Streaming data analysis,
2. High dimensionality,
3. Models scalability.

- To improve

1. Formulation of data abstractions,
2. Distributed computing,
3. Semantic indexing,
4. Data tagging,
5. Information retrieval,
6. Criteria selection for extracting good data representations, and domain adaptation.

4.3. Incremental and ensemble learning

Incremental learning and ensemble learning constitute two learning dynamic strategies. Basic methods in learning from big stream data with concept drift [22] are available.

Incremental and ensemble learning are often useful to data streams and big data. They tackle various difficulties such as addressing data availability, limited resources. They are adapted to many applications such as stock trend prediction and user profiling. Applying incremental learning enable to produce faster classification or forecasting times while receiving new data.

Table 2 shows the machine learning algorithms which uses the incremental learning.

Table 2: Traditional machine learning algorithms

1	Decision trees
2	Decision rule
3	Neuronal networks
4	Gaussian RBF network
5	Incremental SVM

When comparing those types of algorithms, it is noticed that incremental algorithms are faster. On the other hand, ensemble algorithms are more elastic and can get used to concept drift. Furthermore, we have to remember that;

1. All classification algorithms cannot be used in incremental learning,
2. Almost every classification algorithms can be used in ensemble algorithms [22].
3. An incremental algorithm can be use in the absence of concept drift or if the concept drift is smooth.
4. Ensemble algorithms are suggested in the case to ensure accuracy in the case of huge concept drift or abrupt concept drift.
5. To deal with relatively simple data-stream or a high level of real-time processing, incremental learning is more suitable.
6. Ensemble learning constitutes a better choice in case of complicated or unknown distribution of data streams.

4.4. Granular computing

Granular Computing (GrC) [23] is not new, but it has recently become more popular for its use in various Big Data domains.

Advantages of GrC in

- Intelligent data analysis,
- Pattern recognition,
- Machine learning
- Uncertain reasoning for huge size of data sets.
- Design of decision making models

GrC comprise a universal computation theory based on granules such as;

1. Classes,
2. Clusters,
3. Subsets,
4. Groups
5. Intervals.

GrC useful in following areas;

To build an efficient computational model for complex Big Data applications such as;

1. Data mining,
2. Document analysis,
3. Financial gaming,
4. Organization and retrieval of huge data bases of multimedia, medical data, remote sensing, biometrics.

Distributed systems require supporting different users in understanding;

1. Big data at different granularity levels.
2. To analyze data and present results with different viewpoints.

GrC can achieve above with powerful tools for multiple granularity and multiple viewing of data analysis. Moreover, GrC techniques can serve as effective processing tools for real world intelligent systems and dynamic environment like FDS (Fuzzy Dynamic Decision Systems).GrC enables to tackle the complex issue of evolving attributes and objects in streams over time.

1. GrC can be useful in research to develop efficient decision-making models dedicated to resolve complex problems of Big Data.
2. GrC techniques can improve the current big data techniques while tackling big data challenges.

5. Big Data and Hadoop Ecosystem

5.1. Hadoop potentials

Apache Hadoop is very famous and widely used a Big Data technology. It helps in to avoid the low performance and the complication comes across when processing and analyzing Big Data using traditional technologies.

The power of Hadoop platform is based on;

1. The Hadoop Distributed File System (HDFS)
2. The MapReduce framework

In addition, users can add modules on top of Hadoop as needed according to their objectives as well as their application requirements (e.g., capacity, performances, reliability, scalability, security). In fact, Hadoop community has contributed to enrich its ecosystem with several open source modules. In parallel, IT venders provide special enterprise hardening features delivered within Hadoop distributions.

5.2. Data Storage Layer: HDFS and HBase

To store data, Hadoop relies on both its file system HDFS and a non relational database called Apache HBase.

Hadoop Distributed File System (HDFS)

HDFS is a data storage system. It supports up to hundreds of nodes in a cluster and provides a cost-effective and reliable storage capability. It can handle both structured and unstructured data and hold huge volumes (i.e., stored files can be bigger than a terabyte). However, users must be aware that HDFS do not constitute a general purpose file system. This is because HDFS was designed for high-latency operations batch processing. In addition, it does not provide fast record lookup in files. HDFS main advantage is its portability across heterogeneous hardware and software platforms. In addition, HDFS helps to reduce network congestion and increase system performance by moving computations near to data storage. It ensures also data replication for fault-tolerance. Those features explain its wide adoption.

HDFS is based on master slave architecture. It distributes large data across the cluster.

HBase

HBase is a distributed non relational database. This is an open source project that is built on top of HDFS. Important properties of Hbase are ;

1. Suitable for low-latency operations.
2. Based on column-oriented key/value data model.
3. To support high table-update rates and to scale out horizontally in distributed clusters.
4. Provides a easy structured hosting for very large tables in a BigTable-like format.

Tables store data logically in rows and columns. The benefit of such tables is that they can handle dense of rows and dense of columns. HBase permits a lot of elements to be cluster into column families. Thus, HBase is more flexible than relational databases. Instead, HBase has the advantage of allowing users to introduce updates to better handle changing applications requirements. However, HBase has the limitation of not supporting a structured query language like SQL.

Tables of HBase are called HStore and each Hstore has one or more Map-Files stored in HDFS. Each table must have a defined schema with a Primary Key that is used to access the Table. The row is identified by table name and start key while columns may have several versions for the same row key.

Hbase provides many features such us real-time queries, natural language search, consistent access to Big Data sources, linear and modular scalability, automatic and configurable sharding of tables. It is included in many Big Data solutions and data driven websites such as Facebook Messaging Plat-form. HBase includes Zookeeper for coordination services and runs a Zookeeper instance by default. Table 3 summaries the comparisons between HDFC & Hbase.

Table 3: Comparison between HDFS and Hbase features

Properties	HDFS	HBase
System	Distributed file system. Large files can be stored.	Distributed non-relational database. Built on top of HDFS.
Query and search performance	HDFS is not a general purpose file system. It does not provide fast record lookup in files	It enables fast record lookups (and updates) for large tables
Storage	HDFS stores large files (gigabytes to terabytes in size) across Hadoop servers.	HBase internally puts the data in indexed Store Files that exist on HDFS for high-speed lookups
Processing	HDFS is suitable for High Latency operations batch processing	HBase is built for Low Latency operations
Access	Data is primarily accessed through Map Reduce	HBase provides access to single rows from billions of records
Input-ouput operations	HDFS is designed for batch processing and hence does not support random reads/writes operations	HBase enables reads/writes operations. Shell command programming, client APIs using JAVA, REST, Thrift can be used for information access.

5.3. Data Processing Layer

MapReduce and YARN constitute two options to carry out data processing on Hadoop. They are designed to manage job scheduling, resources and the cluster. It is worth noticing that YARN is more generic than MapReduce.

MapReduce programming model

It is one of the First essential steps for the new generation of Big Data management and analytics tools. MapReduce has an interesting benefit for Big data applications. In fact, it simplifies the processing of massive volumes of data through its efficient and cost-effective mechanisms. It enables to write programs that can support parallel processing.

In fact, MapReduce programming model uses two subsequent functions that handle data computations: the Map function and the Reduce function.

More precisely, a MapReduce program relies on the following operations:

1. First, the Map function divides the input data (e.g., long text file) into independent data partitions that constitute key-value pairs.
2. Then, the MapReduce framework sent all the key-value pairs into the Mapper that processes each of them individually, throughout several parallel map tasks across the cluster. Each data partition is assigned to a unique compute node. The Mapper gives outputs as a one or more middle keyvalue pairs. At this stage, the framework is charged to collect all the middle keyvalue pairs, to sort and cluster them by key. So the result is many keys with a list of all the associated values.
3. Next, the Reduce function is used to process the intermediate output data. For each unique key, the Reduce function aggregates the values associated to the key according to a predefined program (i.e., filtering, summarizing, sorting, hashing, taking average or Finding the maximum). After that, it produces one or more output keyvalue pairs.
4. Finally, the MapReduce framework stores all the output keyvalue pairs in an output file.

YARN

YARN is more generic than MapReduce. It provides a better scalability, enhanced parallelism and advanced resource management in comparison to MapReduce. It offers operating system functions for Big Data analytical applications. Hadoop architecture has been changed to incorporate YARN Resource Manager. In general, YARN works on the top of HDFS. This position enables the parallel execution of multiple applications. It allows also handling both batch processing and real-time interactive processing. YARN is compatible with Application Programming Interface (API) of MapReduce. In fact, users have just to recompile MapReduce jobs in order to run them on YARN.

Unlike MapReduce, YARN enhances efficiency by splitting the two main functionalities of the JobTracker into two separate daemons: (1) ResourceManager (RM) that allocates and manages resources across the cluster. (2) Application Master (AM) framework with a library. It is designed to schedule tasks, to match them with TaskTrackers and to monitor their progress. AM negotiates also resources with RM and Node Manager. For instance, it ensures task bookkeeping, maintains counters, restarts failed or slow tasks. Thus, Job scheduling entity ensures lifecycle management of all applications executed in a cluster.

Cascading: a MapReduce framework for complex flows

Cascading framework [24] is a rich Java API that provides many components for fast and cost-effective Big Data application development, testing and integration. Cascading has interesting benefits. It allows managing advanced queries and handling complex workflows on Hadoop clusters. It supports scalability, portability, integration and test-driven development.

This API adds an abstraction level on the top of Hadoop to simplify complex queries through a cascading concept. In fact, the loaded data are processed and split by a series of functions to get multiple streams called flows. Those flows form acyclic-directed graphs and can be joined together as needed.

The pipe assembly defines the flow to run between the data sources (Source Taps) and the output data (Sink Taps) that are connected to the pipe. A pipe assembly may contain one or more Tuples of a given size.

A cascading flow is written in Java and transformed during the execution into classic MapReduce jobs. Flows are executed on Hadoop clusters and are based on the following process:

A Flow instance is a workflow that First reads the input data from one or many Source Taps, and then processes them by executing a collection of parallel or sequential operations as defined by the pipe assembly. Then, it writes the output data into one or several Sink Taps.

A Tuple represents a set of values (like a database record of SQL table) that can be indexed with Fields and can be stored directly into any Hadoop File format as key/value pair. A tuple should have comparable types in order to facilitate Tuple comparison.

Many extensions were added to the Cascading framework to enhance its capabilities, including [25]:

Pattern: used to build predictive big data applications. It provides many machine learning algorithms and enables translating Predictive Model Markup Language (PMML) documents into applications on Hadoop.

Scalding: used as a dynamic programming language to solve functional problems. It is based on Scala language with a simple syntax. This extension is built and maintained by Twitter.

Cascalog: allows to develop application using Java or Clojure (a dynamic programming language based on Lisp dialect). It supports Ad-hoc queries, by running a series of multiple MapReduce jobs to analyze different sources (HDFS, databases and local data). It provides higher level of abstraction than Hive or Pig

Lingual: provides an ANSI-SQL interface for Apache Hadoop and supports a rapid migration of data and workloads to and from Hadoop. Through Lingual, it is easier to integrate the existing Business Intelligence tools and other applications.

Data Querying Layer: Pig, JAQL and Hive

Apache Pig [24] is an open source structure that produces a high level scripting language called Pig Latin. It reduces MapReduce complexity by supporting parallel execution of MapReduce jobs and workflows on Hadoop. Through its interactive environment, Pig like Hive, simplifies exploring and processing in parallel massive data sets using HDFS (e.g., complex data flow for ETL, various data analysis). Pig allows also interaction with external programs like shell scripts, binaries, and other programming languages. Pig has its own data model called Map Data (a map is a set of key-value pairs).

Pig Latin has many advantages. It is based on an intuitive syntax to support an easy development of MapReduce jobs and workflows (simple or nested flows). It reduces the development time while supporting parallelism. Thus, users can rely on Pig Latin language and several operators to upload and process data. Pig Latin is an alternative to Java programming language with scripts similar to a Directed Acyclic Graph (DAG). In fact, in such DAC, operators that process data constitute nodes while data flows are presented by edges. On the contrary to SQL, Pig does not require a schema and can process semi-structured and unstructured data. It supports more data formats than Hive. Pig can run on both the local environment in a single JVM and the distributed environment on a Hadoop cluster.

JAQL [26] is a declarative language above Hadoop that provides a query language and involved in massive data processing. It converts high level queries into MapReduce jobs. It was designed to query semi-structured data based on JSONs (Java-Script Object Notation) format. However, it can be used to query other data formats as well as many data types. So, JAQL like Pig does not require a data schema. JAQL provides several in-built functions, core operators and I/O adapters. Such features ensure data processing, storing, translating and data converting into JSON format.

Apache Hive is a data warehouse system designed to simplify the use of Apache Hadoop. In contrast to MapReduce, that manages data within files via HDFS, Hive enables to represent data in a structured database that is more familiar for users. In fact, Hives data model is mainly based on tables. Such tables represent HDFS directories and are divided into partitions. Each partition is then divided into buckets.

Moreover, Hive provides a SQL-like language called HiveQL that enable users to access and manipulate Hadoop-based data stored in HDFS or HBase. Therefore, Hive is suitable for many business applications.

Hive is not suitable for real-time transactions. In fact, it is based on a low-latency operations. Like Hadoop, Hive is designed for large scale processing so even small jobs may take minutes. Indeed, HiveQL transparently converts queries (e.g., ad hoc queries, joins, and summarization) into MapReduce jobs that are processed as batch tasks.

Unlike most SQL having schema-on-write feature, Hive has schema-on-read and supports multiple schemas, which defers the application of a schema until you try to read the data. Though the benefit here is that it loads faster, the drawback is that the queries are relatively slower. Hive lacks full SQL support and does not provide row-level inserts, updates or delete. This is where HBase worth investing. Table 4 summaries the comparisons between Hive, Pig, JAQL

Table 4: Hive, Pig and JAQL features

Properties	**Properties**		
	Hive	**Pig**	**Jaql**
Language	HiveQL (SQL-like)	Pig Latin (script-based language)	JAQL
Type of language	Declarative (SQL dialect)	Data flow	Data flow
Data structures	Suited for structured data	Scalar and complex data types	File-based data
Schema	It has tables metadata stored in the database	Schema is optionally defined at runtime	Schema is optional
Data Access	JDBC, ODBC	PigServer	Jaql web server
Developer	Facebook	Yahoo	IBM

5.4. Data Access Layer

Data Ingestion: Sqoop, Flume and Chukwa

Apach Sqoop [27] is an open source software-tool. It provides a command-line interface (CLI) that ensures an efficient transfer of bulky data among Apache Hadoop and structured data stores (such as RDBMS, enterprise data-warehouses and NoSQL databases). Sqoop offers many advantages. For instance, it provides fast performance, fault tolerance and optimal system utilization to reduce processing loads to external systems. The transformation of the imported data is done using MapReduce or any other high-level language like Pig, Hive or JAQL. It allows easy integration with HBase, Hive and Oozie. Sqoop brings in data from HDFS, It stores the output in multiple files. Files are;

1. Delimited text files,
2. Binary Avro or Sequence Files containing serialized data.

Reading, parsing, inserting are the common operations the Sqoop Export can perform with the help of HDFS.

Flume [28] is designed to collect, aggregate and transfer data from external technology to HDFS. It has a easy elastic structural design and handles streaming of information flows. Flume is based on a simple extensible data model to handle massive distributed data sources. Flume provides various features including fault-tolerance, tunable reliability mechanism as well as failure-recovery service. Though that Flume complements well Hadoop, it is an independent component that can work on other platforms. It is known for its capacity to run various processes on a single machine. By using Flume, users can stream data from various and high volume sources (like Avro RPC source and syslog) into sinks (such as HDFS and HBase) for real-time analysis.

Chukwa [29] is a information gathering structure base above of Hadoop. Chukwas goal is to monitor large distributed systems. For collecting data from all data sources, HDFS is preferred. It uses MapReduce to analyze the gathered data. It inherits Hadoop scalability and robustness. It provides an interface to display, monitor and analyze results

Chukwa offers a flexible and powerful platform for Big Data. It enables analysts to collect and analyze Big Data sets as well as to monitor and display results.

Chukwa is based on four main components: First, it relies on data agents on each machine to emit data. Next, collectors are used to collect data from agents and write it to a stable storage. MapReduce jobs are used to parse and archive data. Users can rely on a friendly interface (HICC) to display results and data. It has a web-portal style. Table 5 summarizes the comparisons between Flume & Chukwa.

Table 5: A comparison between Flume and Chukwa

Properties	**Projects**	
	Chukwa	**Flume**
Real-time	Information acquisitions are done periodically and analysis is done real time.	Its center of attention is on constant real-time analysis (in seconds)
Architecture	Batch-system	Continuous stream processing system
Manageability	It distributes information about data flows broadly among its services	Preserve a central record of continuing information floods, stored repeatedly by means of Zookeeper
Reliability	Agent on each node do the task of finalizing which information to send Chukwa employs an end-to-end release model that can influence local on-disk record documents for consistency	Vigorous/error tolerant with tunable dependability methods and failover and improvement mechanisms. Flume takes on a hop-by-hop model.

Data streaming: storm and spark

Storm [24] is an open source distributed system. The advantage of Strom;

1. To handling real time data operations, processing.
2. An easy-to-use
3. Rapid
4. Scalable
5. Fault tolerant.
6. Automatically restart failure process by diverting it to another node.
7. Useful in real time analytics.
8. Online machine learning.

In comparison to flume, Storm shows better efficiency in implementing complex processing requirements by relying on the Trident API.

Storm is based on a topology composed of a complete network of spouts, bolts, and streams.

The interface of Strom is ISpout. This interface can support any type of incoming data. Many system which are synchronous are used to consume the data. This is also applicable to asynchronous system. Examples of such real-time system are;

1. JMS,
2. Kafka,
3. Shell
4. Twitter).

This Storm make it possible to perform the write operations to any output system. Another interface known as IBolt supports any output system.

Examples are ;

1. JDBC
2. Sequence Files,
3. Hadoop HDFS, Hive, HBase, and other messaging system.

Storm is used to prepare results that can then be analyzed by other Hadoop tools. It can process million tuples per second. Like MapReduce, Storm provides a simplified programming model, which hides the complexity of developing distributed applications.

Apache Spark is an open source distributed processing framework that was created at the UC Berkeley AMPLab. Spark is like Hadoop but it is based on in-memory system to improve performance. It is a recognized analytics platform that ensures a fast, easy-to-use and flexible computing. Spark handles complex analysis on large data sets. Indeed, Spark execute the operations very faster than Hive and Apache Hadoop via MapReduce in-memory system. Spark is based on the Apache Hive codebase. In order to improve system performance, Spark swap out the physical execution engine of Hive. In addition, Spark offers APIs to support a fast application development in various languages including Java, Python and Scala. Spark is able to work with all files storage systems that are sup-ported by Hadoop.

Sparks data model [30] is based on the Resilient Distributed Dataset (RDD) abstraction.

1. RDDs comprise a read-only gathering of items stored in system memory from corner to corner in multiple machines.
2. These items are available with no require of a disk access.
3. These items can be rebuilt if a partition is lost.

The Spark can support various functions like;

1. Task scheduling,
2. Memory management,
3. Fault recovery,
4. Interacting with storage systems, etc.

Above functions can be possible with help of following Spark components.

1. **Spark SQL** [30]: One important feature of Spark SQL is that it unifies the two abstractions: relational tables and RDD. So programmers can easily mix SQL commands to query external data sets with complex analytics. Concretely, users can run queries over both imported data from external sources (like Parquet files an Hive Tables) and data stored in existing RDDs. In addition, Spark SQL allows writing RDDs out to Hive tables or Parquet files. It facilitates fast parallel processing of data queries over large distributed data sets for this purpose. It uses a query languages called HiveQL. For a fast application development, Spark has developed the Catalyst framework. This one enable users via Spark SQL to rapidly add new optimizations.
2. **Spark streaming** [31]: Spark Streaming is another component that provides automatic parallelization, as well as scalable and fault-tolerant streaming processing. It enables users to stream tasks by writing batch like processes in Java and Scala. It is possible to integrate batch jobs and interactive queries. It runs each streaming computation as a series of short batch jobs on in-memory data stored in RDDs.
3. **MLlib** [32]: MLlib is a distributed machine learning framework built on top of Spark. For performance, MLlib provides various optimized machine learning algorithms such us classification, regression, clustering, and collaborative filtering. Like Mahout, MLlib is useful for machine learning categories. They offer algorithms for topic modeling and frequent pattern mining. Mlib supports also regression Models. However, Mahout does not support such model. MLlib is relatively young in comparison to Mahout.
4. **GraphX, Wendell2014**: GraphX constitutes a library for manipulating graphs and executing graph-parallel computations. GraphX enlarge the features of Spark RDD API.

GraphX uses;

1. Graphs manipulation (e.g., subgraph and mapVertices).
2. It gives a library of graph algorithms (e.g., PageRank and triangle counting). Table 6 summarizes the comparison between Strom & Spark.

Table 6: A comparison between Strom and Spark

Properties	**Projects**	
	Sprak	**Storm**
Foundation	UC Berkeley	BackType, Twitter
Type	Open source	Open source
Implementation language	Scala	Coljure
Supported languages	Java, Python, R, Scala	Any
Execution model	Batch, streaming	Streaming
Latency	Spark has latency of just few seconds (Deponding on batch size)	Strom has latecy of sub-seconds
Management style	Spark writes data to the storage and requires stateful Computations	Storm rools on it own or uses trident and requires stateless computations
Fault Tolerance	Support only exactly once processing mode	Supports exaclty once,at least once and at most once processing mode
Stream sources	HDFS	Spout
Stream Computation	Windows Operations	Bolts
Stream Primitives	Dstream	Tuple
Provisioning	Basic monitoring using ganglia	Apache Ambari
Resources Manger Integration	Messos and Yarn	Mesos
Hadoop Distr	HDP, CDH, MapR	HDP

Storage Management: HCatalog

Apache HCatalog [33] provides a table and storage management service for Hadoop users. It enables interoperability across data processing tools (like Pig, Hive and MapReduce). This is achieved through a shared schema and data type mechanisms. It provides an interface to simplify read and write data operations for any data format (e.g., RCFile, CSV, JSON and SequenceFiles formats) for which a Hive SerDe (serlializer-deserializer) can be written. For that, The system administrator provides the Input Format, Output Format and the SerDe.

The abstracted table of HCatalog provides a relational view of data in HDFS and allows to view disparate data formats in a tabular format. So users do not have to know where and how data is stored. Furthermore, HCatalog supports users with other services. It notifies data availability and provides a REST interface to permit access to Hive Data Definition Language(DDL) operations [33]. It also provides a notification service that notifies workflow tools (like Oozie) when new data becomes available in the warehouse.

5.5. Data analytics

Apache Mahout [24] is an open source machine learning software library. Mahout can be added on top of Hadoop to execute algorithms via MapReduce. It is designed to work also on other platforms.

Mahout [34] is essentially a set of Java libraries. It has the benefit of ensuring scalable and efficient implementation of large scale machine learning applications and algorithms over large data sets. Indeed, Mahout library provides analytical capabilities and multiple optimized algorithms. For instance, it offers libraries for clustering (like K-means, fuzzy K-means, Mean Shift), classification, collaborative filtering (for predictions and comparisons), frequent pattern mining and text mining (for scanning text and assigning contextual data).

Extra tools helps in operations like;

1. Topic modeling,
2. Dimensionality reduction,
3. Text vectorization,
4. Similarity measures,
5. A math library.

R [35] is a programming language.

R can be used in ;

1. Used for statistical computing,
2. Machine learning and
3. Graphics.

R is free, open-source soft-ware distributed and maintained by the R-project that relies on a community of users, developers and contributors.

R programming language includes;

1. A well-developed, simple and effective functionalities,
2. Conditionals, loops,
3. User-defined recursive functions and input and output facilities.

Many Big Data distributions (like Cloudera, Hortonworks and Oracle) use R to perform analytics.

One drawbacks of R is its limited capacity to handle extremely large datasets because of the one node memory limitations. In fact, R like other high-level languages leads to memory overload because it is based on temporary copies instead of referencing existing objects.

A single thread is used to execute the R programs which stored in RAM. So care should be taken that the database size should not greater than RAM size.

R packages are;

1. ff package
2. big-memory Package
3. snow Package
4. Teradata Aster R which runs on the Teradata Aster Discovery Platform [36],
5. pdDR project [37]

Some of above make it possible to implement high-level distributed data parallelism in R.

R provides a more complete set of classification models (regarding the types and depth of algorithms) in comparison to Mahout [38]. However, R is not a rapid solution when com-pared to other environment because of its object-oriented programming that case memory management problems. Indeed, it may be more practical to use Mahout, Spark, SAS or other frame-works to ensure a better performance of extensive computations.

Ricardo is another eXtreme Analytics Platform (XAP) project of IBM Almaden Research Center. This is designed to handle deep analytics problems. It combines the features of Hadoop with those of R as two integrated partners and components. In fact, Ricardo handles many types of advanced statistical analysis through R functionalities (like K-means, clustering, time-series, SVM classification). It leverages also the parallelism of Hadoop DMS.

Experiments showed that Ricardo improves R performance and facilitates operations such us data exploration, model building, model evaluation over massive data sets. Table 7 summaries the comparison between Apache Mahout & R

5.6. Management layer

Coordination and Workflow: Zookeeper, Avro and Oozie

Zookeeper [39] is an open source service designed to coordinate applications and clusters in Hadoop environment. It provides several benefits.

For instance, Zookeeper sup-ports high performance and data availability. It simplifies also distributed programming and ensures reliable distributed storage. JAVA is used to create it. It provides the API for API and C programs Zookeeper is a distributed application based on a client-server architecture. Zookeepers server can run across several clusters. Zookeeper has a file system structure that mirrors classic file system tree architectures. Through its simple interface, Zoo-keeper enables also to implement fast, scalable and reliable cluster coordination services for distributed systems. For instance, it pro-vides the configuration management service that allows a distributed setup, the naming service to Find machines within large cluster, the replicated synchronization service to protect data and nodes from lost, the locking service that enables a serialized access to a shared resource as well as the automatic system recovery from failures. ZooKeeper is based on an in-memory data management. Thus, it ensures distributed coordination at a high speed. Zoo-keeper is increasingly used within Hadoop to provide high avail-ability for the ResourceManager. It is used also by HBase to ensure servers management, bootstrapping, and coordination.

Table 7: A Comparison between Mahout and R

Properties	**Analytical Tools**	
	Apache Mahout	**R**
Type	Open source	Open source
Programming language	JAVA	R language
Architecture	Mostly MapReduce, porting to spark	In-memory system
Supported platform	All Hadoop distributions and other platforms	Hadoop Cloudera Hortonworks Oracle
Features	Its data model is based on Resilient Distributed Datasets (RDDÕs). APIs for rapid application development). Support SQL, HiveQL and Scala through Spark-SQL. Efficient query execution by lyst framework. High level tools to interact with data. Efficient query execution by Catalyst framework. High level tools to interact with data.	Programming language . Libraries with optimized algorithm for machine learning algorithm and graph.
Key Benefits	New users can get started with common use cases quickly. It translate machine learning task expressed in JAVA into Map reduce job	Limited performance in case of very large data sets (One-node memory) . Supports statistics and machine learning algorithm. Flexibility to develop programs. Package for more options.

Unlike other components, Apache ZooKeeper [40] can be used outside Hadoop platform. ZooKeeper is used by Twitter, Yahoo and other companies within their distributed systems for configuration management, sharding, locking and other purposes. It is used also by In IBMÕs Big Insights and Apache Flume.

Apache Avro is a structure. This is useful for;

1. Modeling,
2. Serializing
3. Making Remote Procedure Calls (RPC) [41].
4. Defines a compact and fast binary data design.
5. Efficient data compression and storages at various nodes of Apache Hadoop.

Programming languages such us Java, Scala, C, C++ and Python support this formats [42].

Transforming data program one program to another is very important property of Avro, this happens within Hadoop, Since data is stored with its schema (self-describing), Avro is compatible with scripting languages. There is a data serialization system at the core of Avro. Avro schemas can contain both simple and complex types. Avro uses JSON as an explicit schema or dynamically generates schemas of the exist-ing Java objects.

Apache Oozie [43] is a workflow scheduler system designed to run and manage jobs in Hadoop clusters. It is a reliable, extensible and scalable management system that can handle efficient execution of large volume of workflows. The work-flow jobs take the form of a Directed Acyclical Graphs (DAGs). Oozie can support various types of Hadoop jobs including MapReduce, Pig, Hive, Sqoop and Distcp jobs[44]. One of the main components of Oozie is the Oozie server. This server is based on two main components: a Workflow Engine that stores and runs different types of workflow jobs, and a Coordinator Engine that runs recurrent workflow jobs triggered by a predefined schedule [45]. Oozie enables to track the execution of the workflows. In fact, users can customize Oozie in order to notify the client about the workflow and execution status via Http callbacks (e.g., workflow is complete, workflow enters or exits an action node). Currently, Oozie supports Derby by default in addition to other databases such us HSQL, MySQL, Oracle and PostgreSQL. Oozie provides a collection of APIs library and a command-line interface (CLI) that is based on a client component.

System Deployment: Ambari, Whirr, BigTop and Hue

Apache Ambari [46] is designed to simplify Hadoop management thanks to an intuitive interface. It supports for;

1. Provisioning,
2. Managing, and
3. Monitoring Apache Hadoop clusters

The interface is based on RESTful APIs.

Ambari supports many Hadoop components such us:

1. HDFS,
2. MapReduce,
3. Hive,
4. HCatalog,
5. HBase,
6. ZooKeeper,
7. Oozie,
8. Pig
9. Sqoop.

Moreover, Ambari ensures security over Hadoop clusters using Kerberos authentication protocol.

Apache Whirr [47] is used for;

1. Simplify the creation and deployment of clusters in cloud environments (e.g. Amazons AWS).
2. It provides a collection of libraries for running cloud services.
3. This is available as a command-line tool.
4. This can use locally or within the cloud.
5. Whirr is used to spin up instances and to deploy and configure Hadoop.

In addition, Apache Whirr supports provisioning of Hadoop as well as Cassandra, ZooKeeper, HBase, Valdemort (key-value storage), and Hama clusters on the cloud environments.

BigTop [48] supports Hadoop ecosystem. It aims to develop packaging and verify Hadoop-related projects such as those developed by the Apache community. The goal is to evaluate and to ensure the integrity and the reliability of the system as a whole rather than to evaluate each sub-module individually.

Hue [49] is a web application for interacting with Hadoop and its ecosystem. Hue [50] is friendly with any edition of Hadoop and is existing in all of the most important Hadoop distributions.

5.7. Hadoop distributions

Several IT companies like IBM, Cloudera, MapR & Hortonworks created distributions.

Objectives are;

1. To guarantee compatibility,
2. Security
3. Performance

Many such distributions give services as;

1. Distributed storage systems.
2. Resource management.
3. Coordination services.
4. Interactive searching tools.
5. Advanced intelligence analysis tools.

Cloudera

Cloudera [51] is one of the widely used Hadoop distributions.
This gives support for;

1. Deploying &
2. Managing an Enterprise Data Hub powered by Hadoop.
3. It helps in structured & unstructured information [52].

It's useful in;

1. A centralized administration tool.
2. A unified batch processing.
3. An interactive SQL.
4. A role-based access control.

Other properties of Cloudera are;

1. It's faster than Hive.
2. Query can execute 10 times faster than Hive as well as then Mapreduce.
3. Real-time responsiveness for HiveQL/MapReduce.

Disadvantages of Cloudera are ;

1. Not suitable for querying streaming data such as streaming video or continuous sensor data.
2. All joins operations are performed in memory are limited by the smallest memory node present in the cluster.
3. Single point failure during query execution.
4. Cloudera Enterprise RTQ does not support internal indexing for files and does not allow to delete individual rows.

Hortonworks Data Platform

The Hortonworks Data Platform (HDP) [53] is above Apache Hadoop.
It's properties are;

1. To handle Big Data storage.
2. Querying.
3. Processing.
4. It's rapid.
5. It's cost-effective.
6. Scalable.
7. Management, monitoring and integration of information integration.
8. Support DHFS.
9. Support Hbase.
10. Support MapReduce
11. Support Hue
12. Support Pig.

Amazon Elastic MapReduce (EMR)

Amazon Elastic MapReduce (Amazon EMR) [54] is a web-based service built on Hadoop framework. It has the benefit of providing an easy, rapid and effective processing of huge data sets. In addition, it allows resizing on demand the Amazon clusters by extending or shrinking resources. Thus, it is possible to easily extract valuable insight from big data sources without caring about the Hadoop complexity.
This solution is popular in many industries and supports different goals such as;

1. Log analysis.
2. Web indexing.
3. Data warehousing.
4. Machine learning.
5. Financial analysis.
6. Scientific simulation.
7. Bioinformatics.

It can handle many data source and types, including click stream logs, scientific data, etc. Another advantage is that users can connect EMR to several tools like S3 for HDFS, backup recovery for HBase, Dynamo support for Hive. It includes many interesting free components such us Pig and Zookeeper.

MapR

MapR [55] is a money-making distribution for Hadoop intended for venture.
MapR properties are;

1. Better reliability.
2. Better performance.
3. Easy to use of Big Data storage.
4. Easy to use Big Data processing.
5. Helps in analysis with machine learning algorithms.
6. MapR does not use HDFS.
7. This is having it personal MapR File Systems (MapR-FS).

IBM InfoSphere BigInsights

IBM InfoSphere BigInsights is designed to simplify the use of Hadoop in the enterprise environment. It has the required potential to fulfill enterprise needs in terms of Big Data storage, processing, advanced analysis and visualization.
The Basic Edition of IBM InfoSphere BigInsights includes;

1. HDFS.
2. Hbase.
3. MapReduce.
4. Hive.
5. Mahout.
6. Oozie.
7. Pig.
8. ZooKeeper.
9. Hue.

IBM InfoSphere BigInsights Enterprise Edition [56] provides additional important services: performance capabilities, reliability feature, built-in resiliency, security management and optimized fault-tolerance. It supports advanced Big Data analysis through adaptive algorithms (e.g., for text processing). In addition, IBM provides a data access layer that can be connected to different data sources (like DB2, Streams, dataStage, JDBC, etc.). This IBM distribution has other advantages: First, the possibility to directly store data streams into BigInsights clusters. Second, it supports real-time analytics on data streams. This is achieved through a sink adapter and a source adapter to read data from clusters. IBM facilitates also visualization through Dashboards and Big Sheets.

GreenPlum's Pivotal HD

Pivotal HD [57] provides advanced database services (HAWQ) with several components, including its own parallel relational database. The platform combines an SQL query engine that provides Massively Parallel Processing (MPP), as well as the power of the Hadoop parallel processing framework. Thus, the Pivotal HD solution can process and analyze disparate large sources with different data formats. The platform is designed to optimize native querying and to ensure dynamic pipelining.

In addition, Hadoop Virtualization Extensions (HVE) tool supports the distribution of the computational work across many virtual servers. Free features are also available for resource and workflow management through Yarn and Zookeeper. To support an easy management and administration, the platform provides a command center to configure, deploy, monitor and manage Big Data applications. For easier data integration, Pivotal HD proposes its own DataLoader besides the open source components Sqoop and Flume.

Oracle Big Data appliance

Oracle Big Data Appliance [58] merges, in a system, the influence of optimized company standards hardware, Oracle software known how to tackle it. As well as the usefulness of Apache Hadoop open source mechanism. Thus, this solution includes the open source distribution of Cloudera CDH and Cloudera Manager. Oracle Big Data Appliance is presented as a complete solution that provides many advantages: scalable storage, distributed computing, convenient user interface, end-to-end administration, easy-to-deploy system and other features. It supports also the management of intensive Big Data projects.

The Oracle appliance [59] lies on the power of the Oracle Exadata Database Machine as well as the Oracle Exalytics Business Intelligence Machine. The data is loaded into the Oracle NoSQL database. It provides Big Data connectors for high-performance and efficient connectivity. It includes also an open source oracle distribution of R to support advanced analysis.

The Oracle Big Data Enterprise can be deployed using Oracle Linux and Oracle Java Hotspot virtual machine Hotspot.

Windows Azure HDInsight

Windows Azure HDInsight [60] is a cloud platform developed by Microsoft and powered by Apache Hadoop framework. It is designed for Big Data management on the cloud to store, process and analysis any type of large data sources. It provides simplicity, convenient management tools, and open source services for Cloud Big Data projects[62-67]. Furthermore, it simplifies the processing and intensive analysis of large data sets in a convenient way. It integrates several Microsoft tools such as Power Pivot, Power View and BI features. Table 8 summaries the comparisons between Cloudera, Hortonworks & MapR.

Table 8: A Cloudera, Hortonworks and MapR features

Properties	**Cloudera**	**Hortonworks**	**MapR**
Founded Year	Mars 2009	June 2011	2009
License	Multiple versions: Open source and Licensed	Open source	Licensed
GUI	Yes	Yes	Yes
Execution environment	Local or Cloud	Local or Cloud	Local or Cloud (Amazon)
Metadata architecture	Centralized	Centralized	Distributed
Replication	Data	Data	Data + metadata
Management tools	Cloudera Manager	Ambari	MapR Control System
File System Access	HDFS, read-only NFS	HDFS, read-only NFS	HDFS, read/write NFS (POSIX)
SQL Support	Impala	Stinger	Drill
Security	Supports default Kerberos based authentication for Hadoop services.	Supports default Kerberos based authentication for Hadoop services	Supports default Kerberos based authentication for Hadoop services
Deployement	Deployement with Whirr toolkit. Complex deployment compared to AWS Hadoop or MapR Hadoop	Deployement with Ambari. Simple Deployment .	Through AWS Management console
Maintenance	The maintenance and upgrade requires efforts. Job schulding is done through Oozie.	A set of operational capabilities that provide visibility of the health of the clusters .	Easy to maintain as cluster is managed through AWS Management Console and AWS toolkit.
Cost	Cloudera Standard is free. Cloudera entreprise version is proprietary, needs to be purchased separately. Costs are applicable based on components and tools adopted	HDP is the only completely open Hadoop data platform available. All solutions in HDP are developed as projects through the Apache Software Foundation. There are no proprietary extension	Billing is done through AWS on hourly basis.

6. Conclusion

Recent Big Data platforms are supported by a variety of processing, analytical tools as well as dynamic visualization. Such platforms enable to extract knowledge and value from complex dynamic environment. They also support decision making through recommendations and automatic detection of anomalies, abnormal behavior or new trends.

In this paper, we have studied Big Data characteristics and deeply discussed the challenges raised by Big Data computing systems. In addition to that, we have explained the value of Big Data mining in several domains. Besides, we have focused on the components and technologies used in each layer of Big Data platforms. Different technologies and distributions have been also compared in terms of their capabilities, advantages and limits. We have also categorized Big Data systems based on their features and services pro-vided to final users. Thus, this paper provides a detailed insight into the architecture, strategies and practices that are currently followed in Big Data computing. In spite of the important developments in Big Data Field, we can notice through our comparison of various technologies that many short comings exist. Most of the time, they are related to adopted architectures and techniques. Efforts can be made in the area of information organizations, area precise tools and policy in order to generate next generation Big Data infrastructures. Hence, technological issues in many Big Data areas can be further studied and constitute an important research topic.

References

[1] Botta, A., de Donato, W., Persico, V., PescapŽ, A., 2016. Integration of cloud computing and internet of things: a survey. Future Gener. Comput. Syst. 56, 684Ð700.

[2] Weiss, R., Zgorski, L., 2012. Obama Administration Unveils Big Data Initiative: Announces 200 Million in New R&D Investments. Office of Science and Technology Policy, Washington, DC.

[3] Chen, M., Mao, S., Zhang, Y., Leung, V.C., 2014b. Big Data: Related Technologies, Challenges and Future Prospects. Springer.

[4] Letouz, E., 2012. Big Data for Development: Challenges & Opportunities. UN Global Pulse.

[5] Purcell, B.M., 2013. Big Data using cloud computing. Holy Family Univ. J. Technol. Res.

[6] Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E., 2015a. Deep learning applications and challenges in big data analytics. J. Big Data 2, 1.

[7] Khan, N., Yaqoob, I., Hashem, I.A.T., Inayat, Z., Mahmoud Ali, W.K., Alam, M., Shiraz, M., Gani, A., 2014. Big data: survey, technologies, opportunities, and challenges. Sci. World J.

[8] Chen, C.P., Zhang, C.-Y., 2014. Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf. Sci. 275, 314Ð347.

[9] Nahar, J., Imam, T., Tickle, K.S., Chen, Y.-P.P., 2013. Computational intelligence for heart disease diagnosis: a medical knowledge driven approach. Expert Syst. App. 40, 96Ð104.

[10] Park, B.-J., Oh, S.-K., Pedrycz, W., 2013. The design of poly-

nomial function-based neural network predictors for detection of software defects. Inf. Sci. 229, 40Ð57.

[11] Zhou, L., 2013. Performance of corporate bankruptcy prediction models on imbalanced dataset: the effect of sampling methods. Knowledge-Based Syst. 41, 16Ð25.

[12] Yu, H., Ni, J., Zhao, J., 2013. Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced dna microarray data. Neurocomputing 101, 309Ð318.

[13] Di Martino, B., Aversa, R., Cretella, G., Esposito, A., Kołodziej, J., 2014. Big data (lost) in the cloud. Int. J. Big Data Intell. 1, 3Ð17.

[14] Wang, S., Yao, X., 2012. Multiclass imbalance problems: analysis and potential solutions. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 42, 1119Ð1130

[15] Zhou, L., Wang, Q., Fujita, H., 2017. One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies. Inf. Fusion 36, 80Ð89.

[16] Wang, L., 2016. Machine learning in big data. Int. J. Adv. Appl. Sci. 4, 117Ð123.

[17] Tsai, C.-W., Lai, C.-F., Chao, H.-C., Vasilakos, A.V., 2016. Big data analytics. In: Big Data Technologies and Applications. Springer, pp. 13Ð52.

[18] Bishop, C.M., 2006. Pattern recognition. Mach. Learn. 128, 1Ð58.

[19] Jadhav, A., Deshpande, L., 2016. A survey on approaches to efficient classification of data streams using concept drift. Int. J. 4.

[20] Sun, J., Fujita, H., Chen, P., Li, H., 2017. Dynamicfinancialdistress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble. Knowledge-Based Syst. 120, 4Ð14.

[21] Razzak, M.I., Naz, S., Zaib, A., 2017. Deep learning for medical image processing: Overview, challenges and future. arXiv preprint arXiv:1704.06825.

[22] Zang, W., Zhang, P., Zhou, C., Guo, L., 2014. Comparative study between incremental and ensemble learning on data streams: case study. J. Big Data 1, 1Ð16.

[23] Skowron, A., Jankowski, A., Dutta, S., 2016. Interactive granular computing. Granular. Computing 1, 95Ð113.

[24] Mazumder, S., 2016. Big data tools and platforms. In: Big Data Concepts, Theories, and Applications. Springer, pp. 29Ð128.

[25] Nathan, P., 2013. Enterprise Data Workßows with Cascading. OÕReilly Media Inc..

[26] Beyer, K.S., Ercegovac, V., Gomulka, R., Balmin, A., Eltabakh, M., Kanne, C.-C., Ozcan, F., Shekita, E.J., 2011. Jaql: a scripting language for large scale semistructured data analysis. In: Proceedings of VLDB Conference.

[27] Vohra, D., 2016. Using apache sqoop. In: Pro Docker. Springer, pp. 151Ð183.

[28] Hoffman, S., 2015. Apache Flume: Distributed Log Collection for Hadoop. Packt Publishing Ltd..

[29] Shireesha, R., Bhutada, S., 2016. A study of tools, techniques, and trends for big data analytics. IJACTA 4, 152Ð158.

[30] Sakr, S., 2016b. General-purpose big data processing systems. In: Big Data 2.0 Processing Systems. Springer, pp. 15Ð39.

[31] Azarmi, B., 2016b. Scalable Big Data Architecture. Springer.

[32] [32] Landset, S., Khoshgoftaar, T.M., Richter, A.N., Hasanin, T., 2015. A survey of open source tools for machine learning with big data in the hadoop ecosystem. J. Big Data 2, 1.

[33] Wadkar, S., Siddalingaiah, M., 2014b. Hcatalog and hadoop in the enterprise. In: Pro Apache Hadoop. Springer, pp. 271Ð282.

[34] Dinsmore, T.W., 2016. Streaming analytics. In: Disruptive Analytics. Springer, pp. 117Ð144.

[35] Team, R.C., 2000. R Language DeÞnition. R foundation for statistical computing, Austria.

[36] Brown, M.S., 2014. Data Discovery For Dummies, Teradata Special Edition. John Wiley & Sons Inc..

[37] Raim, A.M., 2013. Introduction to Distributed Computing with pbdR at the UMBC High Performance Computing Facility. Technical Report HPCF-2013-2, UMBC High Performance Computing Facility, University of Maryland, Baltimore County.

[38] Ames, A., Abbey, R., Thompson, W., 2013. Big Data Analytics Benchmarking SAS, R, and Mahout. SAS Technical Paper.

[39] Lublinsky, B., Smith, K.T., Yakubovich, A., 2013. Professional Hadoop Solutions. John Wiley & Sons.

[40] Junqueira, F., Reed, B., 2013. ZooKeeper: Distributed Process Coordination. Reilly Media Inc.

[41] Shapira, G., Seidman, J., Malaska, T., Grover, M., 2015. Hadoop Application Architectures. OÕReilly Media Inc..

[42] Maeda, K., 2012. Comparative survey of object serialization techniques and the programming supports. J. Commun. Comput. 9, 920Ð928.

[43] Islam, M.K., Srinivasan, A., 2015. Apache ozie: The Workßow Scheduler for Hadoop. Reilly Media Inc..

[44] Kamrul Islam, M., Srinivasan, A., 2014. Apache Oozie The Workßow Scheduler for Hadoop. OÕReilly Media Inc.

[45] White, T., 2012. Hadoop: The Definitive Guide. Reilly Media Inc..

[46] Wadkar, S., Siddalingaiah, M., 2014a. Apache Ambari. In: Pro Apache Hadoop. Springer, pp. 399Ð401.

[47] Sammer, E., 2012. Hadoop Operations. Reilly Media Inc..

[48] Lovalekar, S., 2014. Big Data: an emerging trend in future. Int. J. Comput. Sci. Inf. Technol. 5.

[49] Chullipparambil, C.P., 2016. Big Data Analytics Using Hadoop Tools (Ph.D. thesis). San Diego State University.

[50] Landset, S., Khoshgoftaar, T.M., Richter, A.N., Hasanin, T., 2015. A survey of open source tools for machine learning with big data in the hadoop ecosystem. J. Big Data 2, 1.

[51] Azarmi, B., 2016b. Scalable Big Data Architecture. Springer.

[52] Prasad, B.R., Agarwal, S., 2016. Comparative study of big data computing and storage tools: a review. Int. J. Database Theory App. 9, 45Ð66.

[53] Azarmi, B., 2016a. The big (data) problem. In: Scalable Big Data Architecture. Springer, pp. 1Ð16.

[54] Sakr, S., 2016. Big data 2.0 processing systems: a survey. Springer Briefs in Computer Science.

[55] Kobielus, J.G., 2012. The forrester wave: Enterprise hadoop solutions, q1 2012. Forrester

[56] Zikopoulos, P., Parasuraman, K., Deutsch, T., Giles, J., Corrigan, D., et al., 2012. Harness the Power of Big Data The IBM Big Data Platform. McGraw Hill Professional.

[57] Hurwitz, J., Nugent, A., Halper, F., Kaufman, M., 2013. Big Data for Dummies. (1st ed.). For Dummies

[58] Dijcks, J.P., 2012. Oracle: Big Data for the Enterprise. Oracle White Paper. Dimiduk, N., Khurana, A., Ryan, M.H., Stack, M., 2013. HBase in Action. Manning Shelter Island.

[59] Murthy, B., Goel, M., Lee, A., Granholm, D., Cheung, S., 2011. Oracle Exalytics in- Memory Machine: A brief Introduction.

[60] [60] Nadipalli, R., 2015. HDInsight Essentials. Packt Publishing Ltd..

[61] Ahmed Oussous, Fatima-Zahra Benjelloun Ayoub Ait Lahecen, Samir Belfair,"Big Data technologies :A survey" Journal of King Saud University Ð Computer and Information Sciences 2017.

[62] VARUN TEJA, T. and ASADI, S.S., 2016. An integrated approach for evaluation of environmental impact assessment - A model study. International Journal of Civil Engineering and Technology, 7(6), pp. 650-659.

[63] JAWAHAR, A. and KOTESWARA RAO, S., 2015. Recursive multistage estimator for bearings only passive target tracking in ESM EW systems. Indian Journal of Science and Technology, 8(26),.

[64] ADITYA VARMA, K.V., MANIDEEP, T. and ASADI, S.S., 2016. A critical comparison of quantity estimation for gated community construction project using Traditional method vs Plan swift software: A case study. International Journal of Civil Engineering and Technology, 7(6), pp. 707-713.

[65] MURALI, A., KAKARLA, H.K. and VENKAT REDDY, D., 2016. Integrating FPGAs with trigger circuitry core system insertions for observability in debugging process. Journal of Engineering and Applied Sciences, 11(12), pp. 2643-2650.

[66] BALA GOPAL, P., HARI KISHORE, K., KALYANA VENKATESH, R.R. and HARINATH MANDALAPU, P., 2015. An FPGA implementation of onchip UART testing with BIST techniques. International Journal of Applied Engineering Research, 10(14), pp. 34047-34051.

[67] BHARADWAJ, M. and KISHORE, H., 2017. Enhanced launch-off-capture testing using BIST design. Journal of Engineering and Applied Sciences, 12(3), pp. 636-643.