International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

An Adaptive Framework for Real-Time Data Analysis

O. E. Emam
Information Systems Department,
Faculty of Computers and Artificial Intelligence,
Helwan University, Cairo, Egypt.

A. Abdo
Information Systems Department,
Faculty of Computers and Artificial Intelligence,
Helwan University, Cairo, Egypt.

A. M. Abd-Elwahab
Business Information Systems Department,
Faculty of Commerce and Business Administration,
Helwan University, Cairo, Egypt.

Abstract— These The primary purpose of this paper is to provide an adaptive framework for real-time data analysis because we live in the digital world, with continuous data streaming and increasing digitization the amount and type of data is growing in amazing speed which is caused by emerging new services as cloud computing, internet of things and location-based services, the era of real-time data processing has arrived.

The continuous stream of data generated by sensors, machines, vehicles, mobile phones, social media networks, and other real-time sources are compelling organizations to imagine what they could do with this data if they could gain insight into it. A real-time streaming platform must meet the needs of data scientists, developers and data center operations teams without requiring extensive custom code or brittle integration of many third-party components. As more and more data are generated and collected, data analysis requires scalable, flexible, and high performing tools to provide insights in a timely fashion. However, organizations are facing a growing big data ecosystem where new tools emerge and "die" very quickly. Therefore, it can be very difficult to keep pace and choose the right tools. So, the most appropriate situation will depend largely on the state of the data to process, how time-bound requirements are determined, and what kind of results we are interested in.

Keywords: Big Data, Stream Processing, Hadoop, Storm, Spark, Flink.

1 INTRODUCTION

Real time big data analytics is referred to the process of analyzing large volume of data at the moment it is produced or used. It is the process of extracting valuable information for the organization using as soon as its stored/created within big data repository/infrastructure [14], [17], [28] and [39].

Real time analytics is a form of big data analytics but rather focus on big data produced/consumed/stored within a live environment. Such as analyzing mass amount of data as it is produced within stock exchanges, banks and branches throughout the globe. It is mostly used in industries/organizations that routinely produce massive amount of data in a very short time. The scope of the analytics can be from multiple sources. It works by fetching/importing big data stored within a system at run time and execute data/big data analysis algorithms over it. The analytics data is delivered to the administrator usually through an analytics software dashboard [2], [13], [16] and [35].

Big data analytics allows enterprises to use large data sets to uncover information about their processes, customers, market and more. Architecture that analyzes and applies big data analytics in real time simply means pushing data through analytics software as it arrives. The combination makes actionable insights faster, allowing a shorter and better decision-making process [1], [2], [10], [22] and [27].

While the term real-time analytics implies practically instant access and use of analytical data, some experts provide a more concrete time frame for what constitutes real-time analytics, such as suggesting that real-time analytics involves data used within one minute of it being entered into the system [3], [21] and [40].

A common example of real-time analytics is a system where managers or others can remotely view order information that's updated as soon as an order is made or processed. By staying connected to an IT architecture, these users will be able to see the orders represented as they happen, therefore tracking orders in real time. Other examples of real-time analytics would be any continually updated or refreshed results about user events by customer, such as page views, website navigation, shopping cart use, or any other kind of online or digital activity. These kinds of data can be extremely important to businesses that want to conduct dynamic analysis and reporting in order to quickly respond to trends in user behavior [4], [7], [11], [23] and [36].

It is critical for modern and agile enterprises to extract information from operational data in real time. Thus, real-time big data extraction proves to be beneficial in boosting revenue, reducing operational costs and enhancing efficiency overall. Real-time big data has numerous advantages for enterprises. It works by helping them make well-informed decisions, unveil new opportunities and gain a new dimension of insight [30] and [39].

Real-time big data is not only about storing petabytes of data in a warehouse. Real-time big data helps by enabling organizations to do something meaningful from the extracted data quickly and cost-effectively. It can help organizations detect fraud while someone is swiping a credit card or placing an order on a website. It works by combining and analyzing the data, thus allowing well-informed decisions to be made at the right time and at the right place. It governs the use of new techniques through which the analysis could be performed on large data sets. All this is possible through various software frameworks associated with real-time big data [12], [28] and [34].

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

Real-time business intelligence (RTBI or Real-Time BI) is the process of sorting and analyzing business operations and data as they occur or are stored. RTBI allows organizations to evaluate business processes and take strategic action on the current overall business environment [6].

RTBI is important in scenarios that require live business insight in a fast-paced environment. RTBI is implemented on operational systems and live data storage components that maintain business processes, events and data in real time. It also works on big data or past data repositories to combine them, derive inferences or compare/correlate previous statistics. RTBI has several types of deployment and operational architectures, including:

- Event-based data analytics that trigger the detection of specific data events.
- Server-less data analytics used to directly extract data from the source, rather than the data warehouse or repository.

This paper *is organized as follows: Section 2 in this paper,* examines *the related work of big data analytics, real-time data processing, frameworks for real-time analytics and a comparative study on big data analytics frameworks. In section 3, the real-time data processing frameworks are discussed. In section 4, an adaptive real-time data analysis framework is proposed. Finally, conclusion and directions for future work are reported in section 5.*

2 RELATED WORK

The previous work can be divided into the following topics: big data analytics [40], [3], [10], [28] and [22], real-time data processing [16], [17], [39] and [35], frameworks for real-time analytics [23], [7], [25], [34], [4] and [38], and a comparative study on big data analytics frameworks [1], [2], [20] and [24].

1.1 BIG DATA ANALYTICS

In [40], Zhu, et al., proposed a five-layer architecture for big data processing and analytics (BDPA), including collection, storage, processing, analytics and application layer. This architecture target to set up a de facto standard for current BDPA solutions, to collect, manage, process, and analyze the vast volume of both static data and online data streams, and make valuable decisions for all types of industries.

In [3], Anjos, et al., proposed the use of hybrid infrastructures such as Cloud and Volunteer Computing for Big Data processing and analysis. In addition, it provided a data distribution model that improved the resource management of Big Data applications in hybrid infrastructures since it supports the reproducibility and predictability of Big Data processing by low and high-scale simulation within Hybrid infrastructures

In [10], Brock and Khan, tried to look at the factors that associated with the usage of big data analytics, by synchronize technology acceptance model (TAM) with organizational learning capabilities (OLC) framework. These models are applied on the construct, intended usage of big data and also the mediation effect of the OLC constructs is assessed.

In [28], Salloum, et al., presented a technical review on big data analytics using Apache Spark. This review focused on the key components, abstractions and features of Apache Spark. More specifically, it shown what Apache Spark has for designing and implementing big data algorithms and pipelines for machine learning, graph analysis and stream processing.

In [22], Iqbal and Soomro, used Apache Storm to analyze and process big data. It explored for companies to understand the Big Data and its notions. In addition, this paper reviewed for the companies to choose between traditional databases and the big data tools. It is empowering the IT managers to think about the Big Data before it is too late.

1.2 REAL-TIME DATA PROCESSING

In [16], Gürcan and Berigel, provided a valuable insight on real-time processing of big data streams, with its lifecycle, tools and tasks and challenges. This paper initially revealed the lifecycle of real-time big data processing, consisting of four phases, that are data ingestion, data processing, analytical data store, and analysis and reporting. In addition, it described tools and tasks of real-time big data processing.

In [17], Gurusamy, et al., focused on one of the essential components of a big data system: processing frameworks with advantages and limitations. These processing frameworks grouped by the state of the data they are designed to handle. Whereas some systems handle data in batches only, while others process data in a continuous stream as it flows into the system and others can handle data in Hybrid processing frameworks.

In [39], Zheng, et al., studied the challenges of big data firstly and concluded all these challenges into six issues. In order to improve the performance of real-time processing of large data, this paper built a kind of real time big data processing (RTDP) architecture based on the cloud computing technology and then proposed the four layers of the architecture, and hierarchical computing model. This paper proposed a multi-level storage model and the LMA-based application deployment method to meet the real-time and heterogeneity requirements of RTDP system.

In [35], Yang, et al., organized a big data real-time processing system based on Strom and other tools, and according to the simulation experiment, the system can be easily applied in practical situation. It can be widely used in many territories such as cloud computing, data mining and so on.

1.3 FRAMEWORKS FOR REAL-TIME ANALYTICS

In [23], Jayanthi and Sumathi, focused on the challenges that real-time stream processing solution addressed using machine learning. Also, this paper analyzed the traditional analytic tools to bridge the gap between data being generated and data that can be analyzed effectively.

In [7], Bartolini and Patella, proposed RAM^{3}S, a framework for the real-time analysis of massive multimedia streams, and applied it on top of three different open source engines for the analysis of streaming Big Data (i.e., Apache Spark, Apache Storm, and Apache Flink).

In [25], Khan, et al., proposed a framework for the dynamic visualization of real time streaming big data, resilient to both its volume and rate of change. Some of the different directions explored include: (a) the efficient processing and consumption

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

of streaming data; (b) the automated detection of relevant changes in the data stream, highlighting entities that merit a detailed analysis; (c) the choice of the best idioms to visualize big data, possibly leading to the development of new visualization idioms; (d) real-time visualization changes.

In [34], Yadranjiaghdam et al., proposed a framework for real-time analysis of Twitter data. This framework is designed to collect, filter, and analyze streams of data and gives us an insight to what is popular during a specific time and condition. The framework consists of three main steps; data ingestion, stream processing, and data visualization components with the Apache Kafka messaging system that is used to perform data ingestion task. Finally, conducted a case study on tweets about the earthquake in Japan and the reactions of people around the world with analysis on the time and origin of the tweets.

In [4], Anjos, et al., presented a framework consisting of composable data-analysis services that can be combined to address needs of specific applications. Also, this paper focused on applications for small and medium-sized organizations, the framework offered a flexible and lightweight approach that allows these organizations to take advantage of Big Data analysis in the cloud infrastructures.

In [38], Zhang, et al., presented an adaptive MapReduce framework designed for an effective use of computational resources in data center networks to deal with real time data intensive applications. it used three methods, feedback control, stochastic learning with smooth filter and kalman filter to implement the framework. in addition to, comparison the workload prediction method and their influence on makespan to reduce the makspan in three different real-world workload scenarios.

2.4 A COMPARATIVE STUDY ON BIG DATA ANALYTICS FRAMEWORKS

In [1], Abuqabita, et al., discussed briefly the most Known analytics framework and categorized it into three classes, Batch analytic, stream analytic and interactive analytics, finally some of dig data challenges have also been presented.

In [2], Alkatheri, et al., analyzed and compared four frameworks, Hadoop, Flink, Spark, and Storm, based on different key performance indicators for measuring their performance. The results of this study shown that Flink performed the best compared to the other frameworks as it achieved the best performance indicators.

In [14], Emam, et al., presented a valuable insight into the understanding of the real-time data analysis frameworks and comparison of popular real-time data processing systems.

In [20], Inoubli, et al., proposed streaming frameworks for Big Data applications to store, analyze and process the continuously captured data. Also discussed the challenges of Big Data and presented an experimental evaluation and a comparative study of the most popular streaming platforms.

In [24], Karakaya, et al., provided comparative results about three stream processing frameworks including: Apache Spark, Flink and Storm. This study modified the YSB in order to get it working in a multi-node environment and provided results about the saturation level of each framework. Also, it measured the resource usage and performance scalability of the frameworks against a varying number of cluster size.

3 REAL-TIME DATA PROCESSING FRAMEWORKS

This section presents an overview of the most popular real-time data processing frameworks. real-time data processing must have strong timeliness which means it must quickly respond to the request from system terminals in a very short time delay. So, at first, real-time big data processing system must have powerful computing ability for big data. A traditional method to process big data is to rely on the powerful computing capabilities of the cloud computing platform to achieve, while for the timeliness it must rely on the ability of the rapid data exchange between system's internal and nodes.

3.1 Hadoop

Apache Hadoop is a collection of open-source software for reliable, scalable, distributed computing that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. The base Apache Hadoop framework is composed of the three main layers as following:

The **first** one is the data storage layer for collecting data, which contains Hadoop Distributed File System (HDFS) – a distributed file system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster nodes.

The **second** layer is the Hadoop Yet Another Resource Negotiator (YARN) – a platform responsible for managing computing resources in clusters and using them for scheduling users' applications. YARN infrastructure, which provides arithmetic resources for job scheduling such as CPU and memory.

The **third** is Hadoop MapReduce – an implementation of the MapReduce programming model for large-scale data processing (software layer) with other processes. MapReduce is Hadoop's native batch processing engine. It's the best framework for processing data in batches.

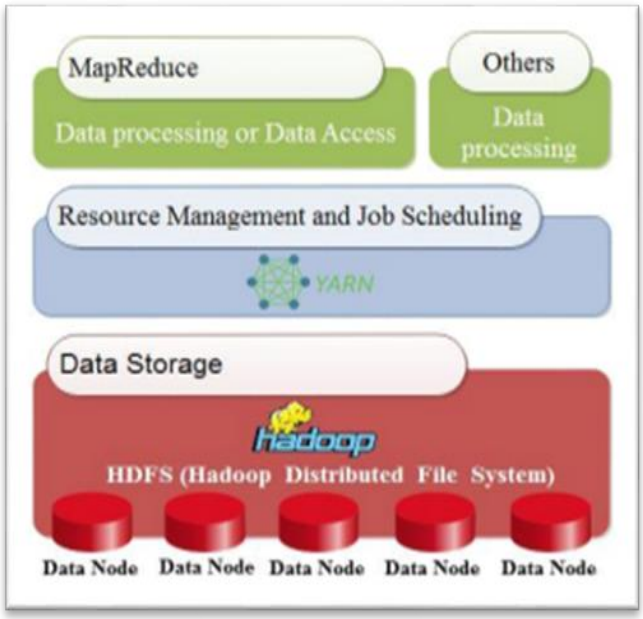

Figure 1. Apache Hadoop Architecture [2]

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

Apache Hadoop and its MapReduce processing engine offer a well-tested batch processing model that is best suited for handling very large datasets where time is not a significant factor. The low cost of components necessary for a well-functioning Hadoop cluster makes this processing inexpensive and effective for many use cases. Compatibility and integration with other frameworks and engines mean that Hadoop can often serve as the foundation for multiple processing workloads using diverse technology [14],[15] and [18].

So, there are numerous companies, enterprise, and organizations utilize Apache Hadoop for two main reasons. First, conducting research for academic or scientific purposes. Second, engaging in the analysis to satisfy customers' needs and help organizations take the right decisions. For example, when the organization needs to know what kind of product customers require. Then, it can produce the product that is needed in abundance, which is one of the several applications of Apache Hadoop.

3.2 Storms

Apache storm is an open-source distributed real-time computational framework that was designed for processing data streams. It is written in Clojure language. Storm focuses on extremely low latency and it can handle very large quantities of data with and deliver results with less latency than other solutions so it's considered one of the best options for workloads that require near real-time processing [22] and [35].

Storm has many use cases: real-time analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Storm is fast: a benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate [24].

Figure 2 shows that a storm process can work with any program language and on any application development platforms. So, it guarantees that data will not be lost.

Figure 2. Apache Storm Architecture [20]

Storm cluster has some similarity with Hadoop. The difference is that its Job in MapReduce running in Hadoop cluster and Topology in Strom. Topology is the highest-level abstract in Storm. Every work process executes a sub-set of a Topology, which consists of multiple Workers running in several machines. But naturally the two frameworks are different. Job in MapReduce is a short-time task and dies with the tasks ending but Topology is a process waiting for a task and it will run all the time as system running unless is killed explicitly.

Figure 3 illustrates the **two** types of nodes: The first is the master node and the second is the worker node. The master node is used for monitoring failures, taking the responsibility of distributed node, and specifying each task for each machine. All these tasks are collectively known as Nimbus, which is similar to Hadoop's Job-Tracker. The worker node is called Supervisor. It works when Nimbus assigns a specific process to it. Thus, each sub-process of a topology works with many distributed machines. Zookeeper plays the role of coordinator between Nimbus and the Supervisors. More importantly, if there is a failure in any cluster, it reassigns the task to another one. So, the slave node controls the execution of its own tasks.

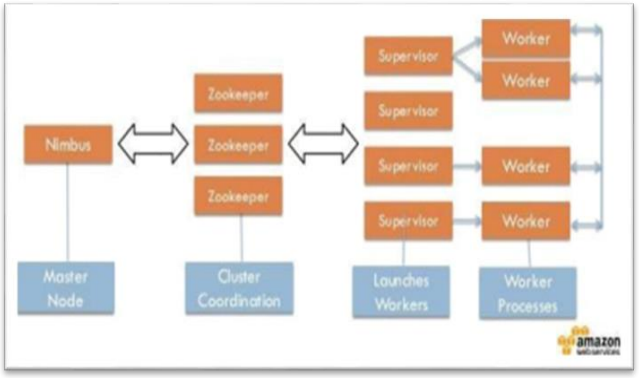

Figure 3. Apache Storm Processing [35]

For pure stream processing workloads with very strict latency requirements, Storm is probably the best mature option. It can guarantee message processing and can be used with a large number of programming languages. Because Storm does not do batch processing, you will have to use additional software if you require those capabilities.

3.3 Spark

Apache Spark is an open-source distributed general-purpose cluster-computing framework that is a unified analytics engine for large-scale data processing and the next generation batch processing framework with stream processing capabilities. Spark framework is to Hadoop what MapReduce is to data processing and HDFS. In addition, Spark has data sharing known as Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG) [14].

Apache Spark achieves high performance for both batch processing workloads by offering full in-memory computation and processing optimization and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine. Spark can run as a standalone or on top of Hadoop YARN, where it can read data directly from HDFS [5] and [8].

Figure 4 represents Spark architecture, which is very easy and fast for selecting a huge amount of data processing. Spark mainly consists of five layers. The first layer comprises of data storage systems such as HDFS and HBASE. The second layer is resource management; for instance, YARN and Mesos. The third is a Spark core engine. The fourth is a library, which is composed of SQL, stream processing, MLlib for machine learning, Spark R, and GraphX for graph processing. The last layer is an application program interface such as Java or Scala. In general, Spark offers a large-scale data processing framework used by banks, telecommunication companies, game companies, governments, and firms such as Apple, Yahoo and Facebook.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

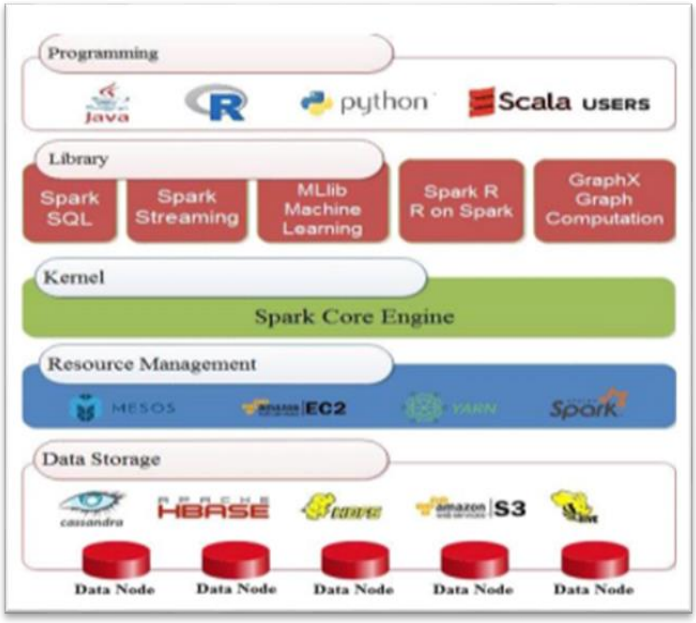

Figure 4. Apache Spark Architecture [23]

A spark is a great option for those with diverse processing workloads. Spark batch processing offers incredible speed advantages, trading off high memory usage. Spark Streaming is a good stream processing solution for workloads that value throughput over latency.

3.4 Flink

Apache Flink is an open-source stream-processing framework and distributed processing engine for stateful computations over unbounded and bounded data streams.

Flink has been designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. It uses in-memory processing technique and provides a number of APIs such as stream processing API (data stream), batch processing API (data set), and table API that has been used for queries. It has machine learning (ML) and graph processing (Gelly) libraries as well. The core of Apache Flink is a distributed streaming dataflow engine written in Java, Scala, Python and SQL and are automatically compiled and optimized into dataflow programs that are executed in a cluster or cloud environment [11]. Flink provides a high-throughput, low-latency streaming engine as well as support for event-time processing and state management. Flink applications are fault-tolerant in the event of machine failure and support exactly-once semantics.

Apache Flink is an excellent choice to develop and run many different types of applications due to its extensive features set. Flink's features include support for stream and batch processing, sophisticated state management, event-time processing semantics, and exactly-once consistency guarantees for state. Moreover, Flink can be deployed on various resource providers such as YARN, Apache Mesos, and Kubernetes but also as stand-alone cluster on bare-metal hardware. Configured for high availability, Flink does not have a single point of failure. Flink has been proven to scale to thousands of cores and terabytes of application state, delivers high throughput and low latency, and powers some of the world's most demanding stream processing applications.

Figure 5 illustrate the architecture of Flink. In the base layer, the storage layer can read and write the data from multiple destinations such as HDFS, local files, and so on. Then, the deployment and resource management layer contain the cluster manager for managing the tasks of planning, monitoring the jobs, and managing the resources. This layer also contains the environment that executes the program, which are the clusters or cloud environments. Besides, it has the local area for single Java virtual machine.

Moreover, it has the kernel layer for distributed stream data flow engine for real-time processing. Also, the application program has interface layers for two processes: batch and streaming. The upper layer is a library in which the program is written in Java or Scala programming language. It is then submitted to the compiler for conversion with the help of the Flink optimizer in order to improve its performance.

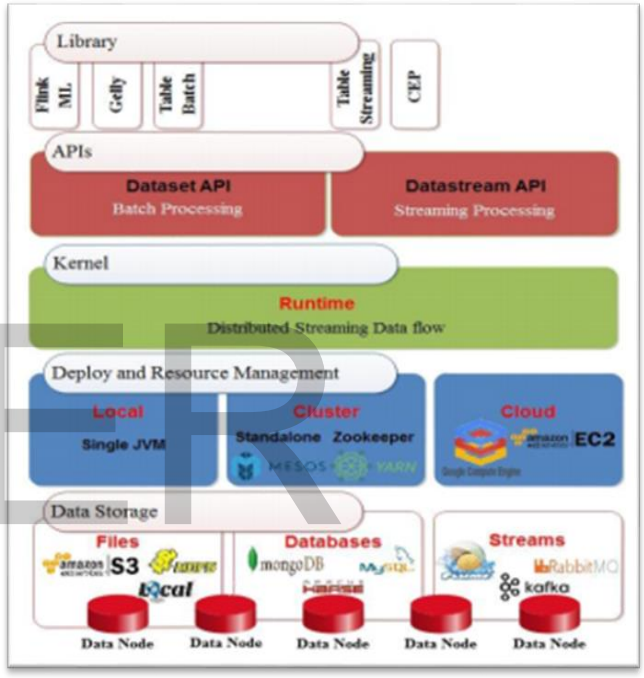

Figure 5. Apache Flink Architecture [2]

Flink offers both low latency stream processing with support for traditional batch tasks. Flink is probably best suited for organizations that have heavy stream processing requirements and some batch-oriented tasks. Its compatibility with native Storm and Hadoop programs, and its ability to run on a YARN-managed cluster can make it easy to evaluate. Its rapid development makes it worth keeping an eye on.

4 AN ADAPTIVE REAL-TIME DATA ANALYSIS FRAMEWORK

In this section, we propose a framework for real-time data analysis according to the demands on the computing ability of real-time data processing system and the timeliness, this framework have five layers: Data, Real-time analytics, Integration, Storage, and Decision-making from a functional level (Shown in Figure 6).

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

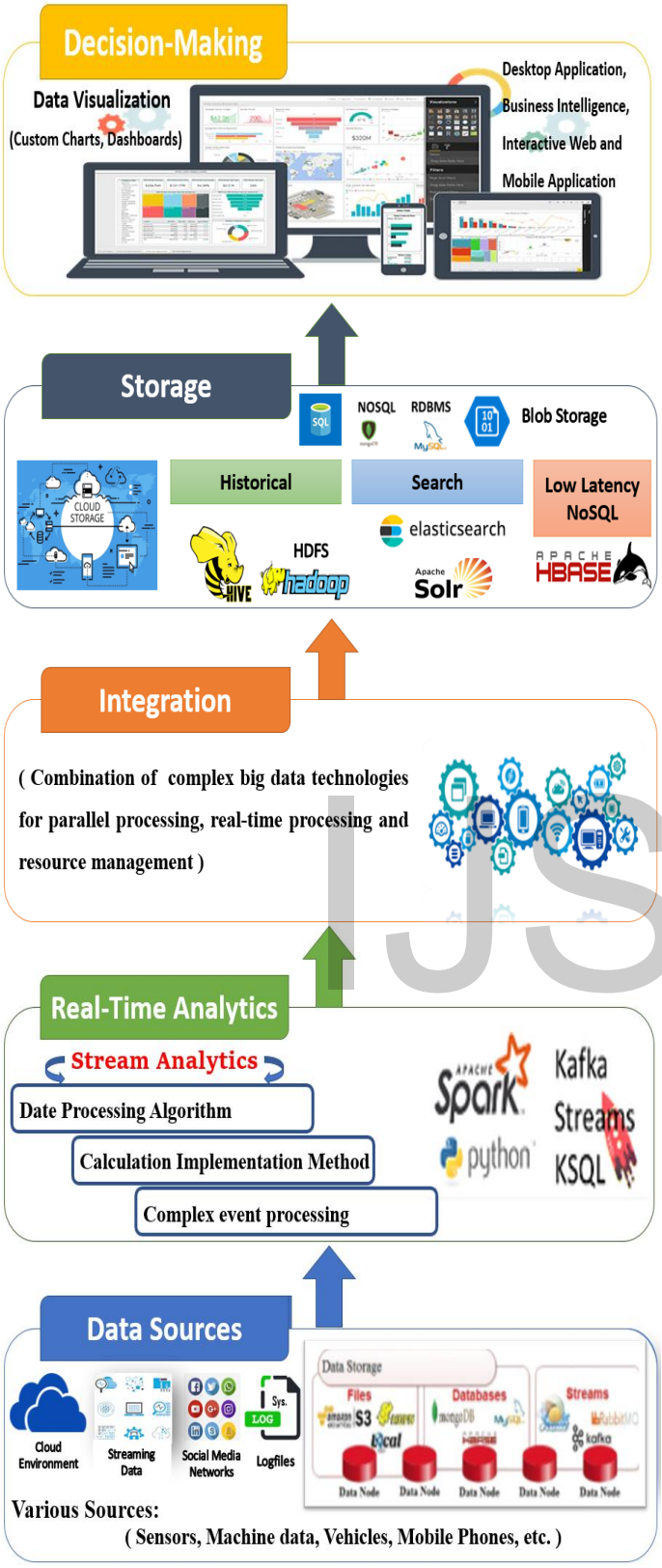

Figure 6. An Adaptive Real-Time Data Analysis Framework

Here, we will discuss each layer in detail from the functionality, processing methods, related tools and deployment aspects of the proposed framework.

4.1 DATA LAYER

Since the data may be generated from systems already in the cloud environment, or external data streaming, or from devices local to a person or piece of equipment or machine data and various sources such as – sensors, social media networks, mobile phones or other sources. Additionally, the events from these systems may be delivered on time as they occur, or batched with timestamps to be delivered when connectivity it available. So, this data is rough and messy, and original data often contain too much useless data, modeling and data analysis for the tremendous difficulties, so the data collection process must be preliminary data analysis and filtering. first need to extract the data features, integrated data sources, extraction points of interest, select the characteristic function to determine the data formats and extract useful information from data marts, and several steps in which the data feature extraction for unstructured text data, etc. of data is very important, therefore, makes the feature extraction for data collection and storage is an important part of the process. This layer mainly responsibilities for data collection and databases, data transmission, data selection and extracting, data storage and cleaning, and preparing data for analytics.

Therefore, RTDP systems can handle data from various data sources, including Hadoop for unstructured storage, the data warehouse system for structured storage and analysis, SQL databases (such as NoSQL, Hbase, or Impala), and some other data source system. Data tools, such as Hive, Apache Storm and Apache Spark, are all useful at this level.

4.1.1 Data Collection and Databases

Data collection refers to the process of retrieving raw data from real-world objects. The process needs to be well signed, allowing specific, structured information to be gathered in a systematic fashion, subsequently enabling data analysis to be performed on the information. Otherwise, inaccurate data collection would impact the subsequent data analysis procedure and ultimately lead to invalid results. As a result, there are many kinds of data collection methods. The **three** common methods for big data collection are: **Log file**, one of the most widely deployed data collection methods, are generated by data source systems to record activities in a specified file format for subsequent analysis. Log files are useful in almost all the applications running on digital devices. **Sensor**, Sensors are used commonly to measure a physical quantity and convert it into a readable digital signal for processing (and possibly storing). Sensor types include acoustic, sound, vibration, automotive, chemical, electric current, weather, pressure, thermal, and proximity. Through **wired** or **wireless networks**, this information can be transferred to a data collection point.

According to the RTDP systems heterogenous platforms, there are a variety of ways to collect data from different data sources and this data can be roughly divided into the CEP, DSMS, DBMS, based on a variety of ways such as MapReduce batch for each treatment have their different data acquisition techniques, such as remote medical field for surgical treatment of complex event processing scenarios for data acquisition **ASIC**, decoding audio and video coding in an **FPGA**, etc. Thus, during the data collection and management there are certain rules that must be collected on the side of the device identification, and can be based on different device programming overhead deployment and management nodes.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

To enhance the data stream processing capabilities of DSMS, can be **pre-distributed** caching and **reuse** method of the intermediate results to avoid each data stream arrives historical repetition processing overhead and makes the data stream localization, **reducing** data between nodes transmission overhead for localized data stream processing, we can use event-driven stage of processing architecture.

4.1.2 Data Storage and Cleaning

In RTDP, data comes from a wide variety of sources, structured and unstructured data mixed. So, Hadoop and other unstructured storage systems in RTDP framework have a natural advantage, but Hadoop itself does not achieve full real-time requirements, which determines our in real-time using Hadoop big data storage process. Hadoop first need to solve real-time problems in the framework of the proposed RTDP use of multi-level storage architecture to solve the problem, its architecture is shown in Figure 7.

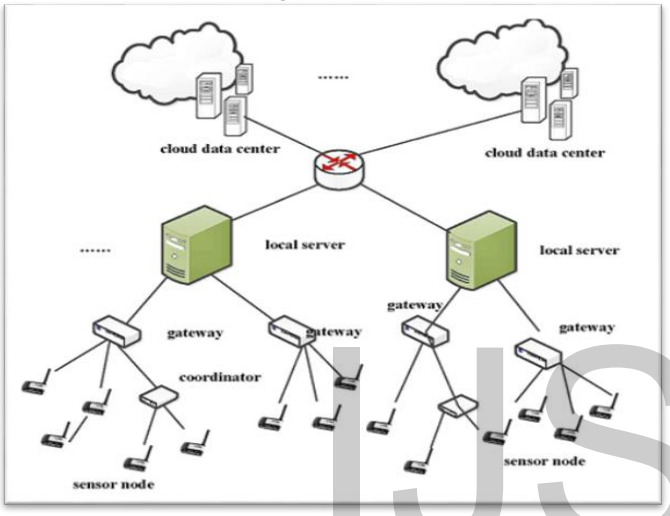

Figure 7. Multi-Level Data Storage Model [39]

In RTDP multi-level storage system data through a lot of the local server first preliminary processing, and then uploaded to the cloud server for in-depth analysis and processing. Such architectural approach to solve the data filtering is how to determine the relevance of the issue of data is an important means, Since the real-time processing of large data nodes need to collect data for rapid processing in the shortest possible time, but also need to filter out unnecessary data, but the data collection process can be verified to collect the current data for post data key input, because data-dependent judgment is a very complex activity.

In RTDP architecture, the local server preliminary data processing, the data collection terminal for rapid response in a very short period of time for the short delay the processing of requests for rapid response, and will not be able to determine the data-dependent data and present without processed data uploaded to the cloud server, the use of cloud computing power for subsequent analysis and processing work due to a limited number of local node, it is generally a PC on the local server capable of working.

Computing resources and the reorganization of the local server can have the cloud computing resources of rationing server, make full use of local and cloud computing capabilities in RTDP architecture because data collection in real time, so multi-level storage system performance bottleneck is the network transmission speed. Use what network to ensure mass real-time data transmission is a major challenge.

4.2 REAL-TIME DATA ANALYTICS LAYER

The analytics layer operates as a production environment for the real-time and dynamic analysis through big data analysis and process, including data characteristic extraction, sample extraction, change of variables, model evaluation, model optimization, and model correction, as with common data warehouse doing big data processing. It also includes a development environment in which developers can construct analytics models for use. Therefore, this layer represents the important stage in the big data value chain.

The main objective of this layer is to extract as much information as possible that is pertinent to the subject under consideration, found a robustness and easily comprehensive prediction model, suggest conclusions and support decision-making. In addition, the feature of RTDP systems instantaneity and big data processing decide that an impeccable RTDP system must be quickly, flexible and with good computing power as well as data reappear. Establishment of a robustness RTDP model is based on comprehensive understanding of needs and on the basis of a comprehensive analysis of the data is based on repeated comparisons of various models, verification basis.

Data analysis is a process for obtaining raw data are collected and converting it into information that is useful for decision-making by users. Data analytics addresses information obtained through observation, measurement, or experiments about a phenomenon of interest. Hence, the nature of the subject and the purpose may vary greatly. The following lists only a few potential purposes:

To extrapolate and interpret the data and determine how to use it.

- To check whether the data are legitimate.
- To give advice and assist decision-making.
- To diagnose and infer reasons for fault, and
- To predict what will occur in the future.

Also, there are different categories of data analytics is as follows:

- **Descriptive Analytics**: exploits historical data to describe what occurred. Descriptive analytics is typically associated with business intelligence or visibility systems.
- **Predictive Analytics**: focuses on predicting future probabilities and trends. For example, predictive modeling uses statistical techniques such as linear and logistic regression to understand trends and predict future outcomes, and data mining extracts patterns to provide insight and forecasts.
- **Prescriptive Analytics**: addresses decision making and efficiency. For example, simulation is used to analyze complex systems to gain insight into system behavior and identify issues and optimization techniques are used to and optimal solutions under given constraints.

This layer mainly responsibilities for guarantee the flexibility and strength of a RTDP system, where tasks in the RTDP framework are controlled prior according to time requirements, tasks with lowest time delay requirements have the highest priority, and the priority can be adjusted during real time process.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

4.2.1 Date Processing Algorithm

Alongside architectural patterns, algorithm framework also plays an important role in RTDP systems computation results. In recent years, numerous scholars have done a lot of research in big data processing algorithm. In the data processing algorithm, condition attribute and decision attribute equivalent matrices merged into one matrix, thus greatly lowered the scale of the equivalent matrices. Moreover, it transformed big data set into serial carry chain computing processes in many subsystems in the computing process, reflected the divide and rule ideology in artificial intelligence field with good practicability and high efficiency. In order to solve the knowledge acquisition problem in big data, there are many scholars have proposed an incremental knowledge acquisition inductive learning algorithm, which to some extent adapted big data knowledge acquisition problem. But MapReduce Algorithm proposed by google pushed big data process into application time.

According to real-time requirements of data processing in RTDP system, analyze of data should be divided into local processing and cloud processing. Among them the local server dispose data collected in data acquisition port, it mainly does basic operation such as data cleansing and data structured analysis. While the cloud server conducts big data analysis and process, offer technical support to decision-making. Data Collection Terminal firstly preprocess data and then submit it to server and deploy the corresponding processing program to local server. The local server then does the Map operation according to data format, then data in heterogeneous nodes will be mapped in different servers, thus avoid data matching problem. The server nodes do the Reduce operation to the data that have been mapped in local servers, then return results to the output terminal. In order to guarantee veracity in data processing, the system supports rolling back action when abnormal thing happens.

Because data processing on local servers are classified according to different types, real-time data processing can be supported in RTDP system. Local computed results update to cloud servers, and make up computed results generated by MapReduce. Thus, not only take advantage of the great computing power of cloud computing, but also guaranteed the instantaneity of data processing.

4.2.2 Calculation Implementation Method

Previously, a two layers calculation mode has been proposed, first, the local server choose local node management and calculation procedures on the local node management, and simple data cleansing and structured modeling. Unstructured data collected by data collector will be transformed into structured data and then uploaded to cloud memory systems and mapped to different management servers. Superstardom makes use of the computing power of cloud terminal to carry through real time computation and analyze.

In recent years, numerous scholars have done a lot of research to the streaming data processing systems, and have made great achievements. Such as, in using an FPGA for video encoding and decoding processing can be used during the data collection terminal DSMS for analysis and filtering, in this mode corresponds to an FPGA acquisition DSMS in a Node, DSMS FPGA collected a large number of data flows flow cytometric analysis performed at the same time to upload data to the cloud server, cloud server will each FPGA configuration tasks performed Reduce Node operation, the calculation results compiled through the LMA choose the right way back to the data collector.

4.2.3 Complex event processing

After determining the mode of calculation methods we also have to realize the model task allocation, as well as prioritization and other operations to make detailed and accurate control, so it relates to a number of complex event processing issues that is, how efficiently a plurality of basic events complex compound has a more complex semantic events, including consideration of constraints between events, and even in some applications to continue to detect complex event to generate a higher level of complex event.

Complex event processing has arisen in recent years as the preferred method for leveraging streaming data. Technologies for simple event processing have been available for years, but most are designed to monitor only one stream of events at a time. Even if users monitor multiple streams, they end up with multiple, siloed views into real-time business operations. The newer practice of CEP can monitor multiple streams at once while correlating across multiple streams, correlating streaming data with data of other vintages, and continuously analyzing the results. Single-purpose, standalone CEP tools are available from a handful of vendors today.

Some active database related work discussed for the basic model of the composite event detection, including: a model based on finite automata, Petri net-based model, based on matching tree model and a model based on directed graph of these CEP model is the basic model of the problem, compared with other methods CEP technology has a tense, relevance, semantic richness, heterogeneity and mass and other characteristics, here we will have a brief introduction to some the basic models of CEP issues.

(1) **Automaton model**. This model is used to implement event expression of complex events with regular expressions have a similar form, and spatial interactions and events have a causal relationship with the local power grid model, with a strong ability of temporal evolution. Composite event in any one basic event arrives, the automatic machine will transition from a state to the next state, when the automaton enters an acceptable state, then the composite event has occurred because of the simple automata model is not reversible, some of the basic events in the match had not re-visit after the event, so if event and time to consider the link between the values, the need to introduce additional data structure to hold the time information , then extended to form the automaton model. Additionally, the automatic machine during the transition can be added in the transition predicate more complex numerical limits on the time or conditions to design some special automaton model for the application of certain situations.

(2) **Petri net model**. A Petri net, also known as a place/transition (PT) net, is one of several mathematical modeling languages for the description of parallel and distributed systems. It is a class of discrete event dynamic system. A Petri net is a directed bipartite graph, in which the nodes represent transitions and places. Petri gateway Note interval endpoints because the calculation and reasoning, so Key Petri net and testing complex event which represents basic event input position, the output location represents the composite event, the event represents a composite intermediate calculation process by entering the Token calculated Warp guard function that computes Warp is raised up and mark the position of the node, marking the last node in the sequence occurs when the composite event detection mechanism is through its incremental marked Petri net description of the position. Active database system SAMOS and monitoring systems HiFi both used this model.

(3) **Matching tree model**. Based on tree matching technology is mainly by matching tree of the structure to achieve complex event filtering, basic event as matching tree leaf node levels as a composite event matching intermediate node tree root corresponding nodes are filtered out of the composite of the matching tree root means to achieve a complex event detection. READY and Yeast systems use this technique.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

(4) **Directed graph model.** Directed graph model is similar to matching tree model; a directed graph model uses a directed acyclic graph (DAG) represents the composite event. Nodes are used to describe events; edges represent the synthesis of the event rules. Penetration zero for the event input node of the case, the node is zero the output of said composite of intermediate nodes for each level of composite events tag node can also be simply described composition rules, the node event occurs, node rule is triggered. Sentinel system and the EVE system used this model.

According to the above analysis, based on automata and Petri Nets composite event detection is only matched by event order of arrival, and tree-based or graph filtering do not consider the basic sequence of events or the timing due to a number of events may occur path, the first event did not occur in the case it is possible to filter a second time, resulting in unnecessary overhead, so in practice have some limitations. Also, these two methods are is the basic event filtering, and composite event detection as different steps to deal with, without taking into conducting composite event detection but also the need for basic event filtering conditions. In CompAS systematic basis, considering the basic events and for the integration of complex event processing by the detection complex event while selectively filtering basic events, reducing the response delay of the composite event detection, so this method could become a complex event processing system for RTDP in the future [39].

4.3 INTEGRATION LAYER

This layer plays a connecting role in RTDP system. In this layer it combines a lot of common algorithm packages for data processing. Depending on the scene it calls the appropriate algorithm for data analysis and data display, provides technical support for Analysis layer and at the same time provides a decision support and theoretical basis for Decision layer. Meanwhile the layer also needs to identify the device in data collection layer according to the rules been set and deploys applications.

The integration layer is a connecting link between the preceding and the following layers in RTDP system, the task of this layer is completed under the deployment of the rule engine, on the one hand algorithms for the data analysis layer provides the necessary libraries and algorithm package, requires the integration phase of the algorithm based on data analysis layer needs to provide appropriate resources, advance scheduling and allocation of resources, including the scheduling algorithm on the other hand also need and data access layer interacts LMA underlying data acquisition devices, while access to the cloud through the rules engine application's compiled code, then copy the codes of application program to LMA, and complete the arrangement of application programs.In real-time big data processing system, real-time data processing mechanism is the integration layer and decision-making of the decision, which is the decision-making system administrators and other decision-makers.

Some big data analysis system in real-time decision-making phase and data collection phase systems using the same hardware, but it is different data systems in which a data processing method to the data from the data mart layer varies.

In RTDP system, the network architecture and networking systems that affect system availability and efficiency of the key factors.

Developed rapidly in recent years, mobile AdHoc networks, mesh networks, sensor networks and other new network technology, combined with traditional wired networks, cellular networks, in order to build a large real-time data transmission network to provide the foundation for building real-time transmission network due when not completely abandon the existing legacy network infrastructure, next-generation networks must be a mixture of a variety of network technologies in complex networks. In the future each one has a physical component network module should be able to at any time, any place convenient access to the network.

Now widely used network technology is not designed specifically for RTDP These network technologies are based on the "best effort" thinking, in order to optimize the target point to point connection, the timing of which there are a lot of variability and stochastic behavior, therefore, the real-time high system requirements are often forced to use a dedicated network technologies such as CAN, FlexRay, LIN, MAP and other bus technology, but these networks are limited geographical area network , while in large RTDP systems, many transmission of signals and control commands are global transport require high reliability , so it is necessary to clarify these current network technology in what may be , and how to use the network in large RTDP addition also need to study suitable RTDP new network architecture and networking.

The latest developments in wireless sensor and actuator networks (wireless sensor/actuator network, WSANs) refers to a group of sensors and actuators interconnected via a wireless medium, and can perform distributed sensing and action network. WSANs can observation of the physical world, data processing, data-based decision-making and perform the appropriate action, is considered to be the next one of the key technologies to build RTDP, in constructing RTDP networks play an important role.

4.4 STORAGE LAYER

Generally, solutions that require interactive query response rates across aggregate and filtered data would usually use SQL Database as the destination data storage, whereas solutions as well as data streams may have to be stored or visualized. So, the storage should be on a NoSQL database especially when the results are in different formats such as, tweets or posts that ranging from text to images to videos. Data stored in database may be used later for historical data analysis. However, the value of this type of data usually belongs to the current situation and may be much different in another time and circumstance. while solutions that capture all events for large scale analytics such as, training machine learning models should leverage Blob Storage as the store.

4.5 DECISION-MAKING LAYER

This layer makes decisions with data analysis results which is the highest layer of data processing system as well as the ultimate goal of the process of data analysis. RTDP is a procedure involving iterative interaction of numerous tools and systems.

In fact, decision making layer includes **three** parts of concepts; first, data visualization, second, the test and update the model, and the third is to provide managers for decision making. RTDP system during the process of data processing, with the flow of data, the data at different times with a certain variability, and between data also has a certain relevance.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

therefore change with time and depth data processing, data analysis layer data model created may not meet the current needs, so we need to keep the data processing while the update data and update the data model to adapt to changes in the data on the other hand, decision support layer is the highest level of RTDP system, the purpose is to carry out data processing related decisions, so the layer must visualize the generated output results in order to provide decision-makers to manage related decision-making activities.

4.5.1 Data Visualization

Presenting information is a way that people can consume it effectively. The aim of data visualization is to communicate information clearly and effectively through graphical means.

In general, charts and maps help people understand information easily and quickly. Visualization for big data has become an active research area because it can assist in algorithm design, software development, and customer engagement. Managing larger amounts of data will require faster and more sophisticated databases that are too complex for no specialists to implement. Conversely, existing databases will need extensions to manage the visualization data structures, which are likely to become a standard functionality that database systems will provide.

4.5.2 Model Validation

To ensure that the model is designed to be completely correct, with a certain fault tolerance, and is stable, we need regular model validation tests. The basic objective of model validation is to ensure that the model is really effective to run the model validation phase requires the re-extraction of new data model in an established and validated data after the operation centralized data for comparison. If the model is running correctly, it can be deployed to the production environment. Otherwise, the model needs to be repeated error checking until the model is correct and can be stable operation. So, this part of the model actually contains updates, and two aspects are fixed by bug.

Model is updated in the data flow process according to the data processing requirements of the model update and adjust, and bug fixes occur in the model mismatch under the existing data processing an automated solution is an extremely complex model bug fix task, requiring accurate error positioning, , when necessary, also need systems that can perform a certain degree of auto repair. So, this process is difficult to use mathematical models to express, it has also become a hot research and difficult in the system there are many rules already developed a number of commercial systems for large data run business rules, such as IBM's ILOG real-time analysis using the R language deployment, so how fast the future RTDP system model validation and deployment in to some extent, you can refer to existing business systems models and methods, based on the existing expert system rules engine to quickly build automated modeling and analysis, real-time adjustment of the data, analysis, and model deployment.

4.5.3 Decision Support

Decision support is the ultimate goal of data analysis; in addition, decision support involves the use of a large number of visualization tools to show different measurements for data analysis outcomes. Forms of data presentation including business intelligence systems, desktop office systems and mobile terminal systems, etc. Use of the platform includes applications for data warehousing systems and graphical processing tools.

5 CONCLUSION

This paper proposed an adaptive framework real-time data analysis according to the demands on the computing ability of real-time data processing system and the timeliness. Several details on each of these frameworks along with some of the popular software frameworks such as Hadoop, Storm, Spark and Flink are also provided.

This proposed framework will help the business organizations which are considering big data technology for increasing the value of their business by dealing with them right in the beginning. More specifically, there are a handful of elements essential to real-time data analytics. These include:

- It is most helpful in industries and enterprises that create and deal with a huge amount of data on a daily or weekly basis.
- The framework is designed to collect data the moment when it is run; then the system utilizes big data analytics algorithms to provide insight on the fresh data.
- To handle the large amount of data required for real-time analytics, the process will create batches of data to be sent to and mapped in distinct compute engines; results are then compiled for analysis.
- Big data insights are typically delivered in real time, using an analytics software dashboard, which brings a visual element to the analysis.

REFERENCES

[1] F. Abuqabita, R. Al-Omoush, J. Alwidian1, "*A Comparative Study on Big Data Analytics Frameworks, Data Resources and Challenges*", Modern Applied Science, 13 (7) (2019), 1-14.

[2] S. Alkatheri, S. Abbas, M. A. Siddiqui, "*A Comparative Study of Big Data Frameworks*", International Journal of Computer Science and Information Security (IJCSIS), 17 (1) (2019), 66-73.

[3] J. Anjos, K. Matteussi, P. Souza, C. Geyer, A. Silva Veith, G. Fedak, J. Barbosa, "*Enabling Strategies for Big Data Analytics in Hybrid Infrastructures*", International Conference on High Performance Computing & Simulation (HPCS), Orléans, France, (2018), 869-876.

[4] J. Anjos, M. Assuncao, J. Bez, C. Geyer, E. P. Freitas, A. Carissimi, J. Costa, G. Fedak, F. Freitag, V. Markl, P. Fergus, R. Pereira, "*An Application Framework for Real Time Big Data Analysis on Heterogeneous Cloud Environment*", IEEE International Conference on Computer and Information Technology, Liverpool, United Kingdom (2015), 199-206.

[5] M. Assefi, E. Behravesh, G. Liu, A. P. Tafti, *"Big Data Machine Learning using Apache Spark MLlib",* IEEE International Conference on Big Data, Boston, USA, 1 (11) (2018), 3492–3498.

[6] B. Azvine, Z. Cui, D. Nauck, B. Majeed, "*Real Time Business Intelligence for the Adaptive Enterprise*", International Conference on E-Commerce Technology, San Francisco, CA, USA, (2006), 1-12.

[7] I. Bartolini, M. Patella, "*A General Framework for Real-Time Analysis of Massive Multimedia Streams*", Multimedia Systems, 24 (4) (2018), 1–16.

[8] A. Bhattacharya, S. Bhatnagar, "*Big Data and Apache Spark: A Review*", International Journal of Engineering Research & Science (IJOER), 2 (5) (2016), 206-210.

International Journal of Scientific & Engineering Research, Volume 11, Issue 5, May-2020
ISSN 2229-5518

[9] A. Block, B. Brandenburg, J. H. Anderson, S. Quint, "*An Adaptive Framework for Multiprocessor Real-Time Systems*", Euromicro Conference on Real-Time Systems, Prague, Czech Republic, (2008), 1-12.

[10] V. Brock, H. Khan, "*Big Data Analytics: Does Organizational Factor Matters Impact Technology Acceptance?*", Journal of Big Data, (2017), 4-21.

[11] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "*Apache Flink: Stream and Batch Processing in A Single Engine*", Bull. IEEE Comput. Soc. Tech. Comm. Data Eng., 36 (4) (2015), 28-38.

[12] N. Deshai, S. Venkataramana, B. Sekhar, K. Srinivas,P. S. Singh, L. NagaKrishna, "*An Advanced Comparison on Big Data World Computing Frameworks*", International conference on computer vision and machine learning, IOP Conf. Series: Journal of Physics: Conf. Series 1228 (2019), 1-11.

[13] R. Dugane, A. Raut, "*A Survey on Big Data in Real Time*", International Journal on Recent and Innovation Trend in Computing and Communication, 2 (4) (2014), 794-797.

[14] O. E. Emam, A. Abdo, A. M. Abd-Elwahab, "*A Comparative Study on Real Time Data Analysis Frameworks*", International Journal of Scientific & Engineering Research, 10 (10) (2019), 1059-1065.

[15] R. A. Fadnavis, S. Tabhane, "*Big Data Processing using Hadoop*", International Journal of Computer Science and Information Technology, 6 (1) (2015), 443–445.

[16] F. Gürcan, M. Berigel, "*Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges*", IEEE 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), (2018), 1-6.

[17] V. Gurusamy, S. Kannan, K. Nandhini, "*The Real Time Big Data Processing Framework: Advantages and Limitations*", International Journal of Computer Sciences and Engineering, 5 (12) (2017), 305-311.

[18] A. V. Hazarika, G. J. S. R. Ram, E. Jain, "*Performance comparison of Hadoop and spark engine*", International Conference on IoT in Social, Mobile, Analytics and Cloud (I SMAC), (2017), 671–674.

[19] B. Hiraman, C. Viresh M., K. Abhijeet C., "*A Study of Apache Kafka in Big Data Stream Processing*", International Conference on Information, Communication, Engineering and Technology, India, (2018), 1-3.

[20] W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, E. Nguifo, "*A Comparative Study on Streaming Frameworks for Big Data*", 44th International Conference on Very Large Data Bases: Workshop LADaS - Latin American Data Science, Rio de Janeiro, Brazil, (2018), 1-8.

[21] W. Inoubli, S. Aridhi, H. Mezni, M. Maddauri, E. M. Nguifo, "*An Experimental Survey on Big Data Frameworks*", International Journal of Future Generation Computer Systems-Elsevier, (2018), 1-21.

[22] M. H. Iqbal, T. R. Soomro, "*Big Data Analysis: Apache Storm Perspective*", International Journal of Computer Trends and Technology, 19 (1) (2015), 9–14.

[23] D. Jayanthi, G. Sumathi, "*A Framework for Real-time Streaming Analytics using Machine Learning Approach*", International Journal of Advanced Computer Technology, (2016), 85-91.

[24] Z. Karakaya, A. Yazici, M. Alayyoub, "*A Comparison of Stream Processing Frameworks*", International Conference on Computer and Applications (ICCA), Doha, United Arab Emirates (2017), 1–12.

[25] A. Khan, D. Gonçalves, D. C. Leão, "*Towards an Adaptive Framework for Real-Time Visualization of Streaming Big Data*", Euro graphics International Conference on Visualization (EuroVis), Posters Track, (2017), 1-3.

[26] H. Luu, "*Beginning Apache Spark 2: with Resilient Distributed Datasets, Spark SQL, Structured Streaming and Spark Machine Learning Library*", New York, Apress, (2018).

[27] M. Mittal, V. E. Balas, L. M. Goyal, R. Kumar, "*Big Data Processing using Spark in Cloud*", Springer Singapore, Singapore, (2019).

[28] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "*Big Data Analytics on Apache Spark*", International Journal of Data Science and Analytics, 1 (3–4) (2016), 145–164.

[29] M.A. Srinuvasu, A. Koushik, E.B. Santhosh, "*Big Data: Challenges and Solutions*", *International Journal of Computer Sciences and Engineering*, *5* (10) (2017), 250–255.

[30] J. Veiga, R. R. Expósito, X. C. Pardo, G. L. Taboada, J. Tourifio, "*Performance Evaluation of Big Data Frameworks for Large-Scale Data Analytics*", IEEE International Conference in Big Data, (2016), 424–431.

[31] A. Verma, A. H. Mansuri, N. Jain, "*Big Data Management Processing with Hadoop Mapreduce and Spark Technology: A Comparison*", Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India (2016), 1–4.

[32] J. P. Verma, A. Patel, "*Comparison of MapReduce and Spark Programming Frameworks for Big Data Analytics on HDFS*", International Journal of Computing Science and Communication (IJCSC), 7 (2) (2016), 80-84.

[33] B. Yadranjiaghdam, N. Pool, N. Tabrizi, "*A Survey on Real-Time Big Data Analytics: Applications and Tools*", IEEE International Conference on Computational Science and Computational Intelligence, Greenville, NC, USA (2016), 404-409.

[34] B. Yadranjiaghdam, S. Yasrobi, N. Tabrizi, "*Developing a Real-Time Data Analytics Framework for Twitter Streaming Data*", IEEE 6th International Congress on Big Data, (2017), 329- 336.

[35] W. Yang, X. Liu, L. Zhang, L. T. Yang, "*Big Data Real-Time Processing Based on Storm*", IEEE 12th International Conference on Trust, Security and Privacy in Computing and Communications, (2013), 1784–1787.

[36] A. Yassinea, S. Singh, M. S. Hossain, G. Muhammad. "*IoT Big Data Analytics for Smart Homes with Fog and Cloud Computing*", Future Generation Computer Systems, 91 (2019), 563–573.

[37] Q. Yuan, Z. Feng, F. Jun, M. Qiang, "*Real-Time Processing for High Speed Data Stream Over Large Scale Data",* Chinese Journal of Computers, 35 (3) (2012), 477-490.

[38] F. Zhang, J. Cao, X. Song, H. Cai, C. Wu, "*An Adaptive MapReduce Framework for Real Time Applications*", IEEE 9th International Conference on Grid and Cloud Computing, Nanjing, China, (2010), 1-6.

[39] Z. Zheng, P. Wang, J. Liu, S. Sun, "*Real-Time Big Data Processing Framework: Challenges and Solutions*", International Journal of Applied Mathematics & Information Sciences, 9 (6) (2015), 3169-3190.

[40] J. Y. Zhu, B. Tang, V. O. Li, "*A five-layer architecture for big data processing and analytics*", International Journal of Big Data Intelligence, 6 (1) (2019), 38-49.