Big Data Platforms for Processing Streaming Data

T. Muniyappan[#], K. Saranya[*]
[1]*PG Student, Department of Computer Science*
Bharathidasan University Constituent Arts and Science College
Navalurkuttapattu, Srirangam (Tk), Tiruchirappalli – 620 027, Tamilnadu.
munisabari95@gmail.com
[2]*Faculty, Department of Computer Science*
Bharathidasan University Constituent Arts and Science College
Navalurkuttapattu, Srirangam (Tk), Tiruchirappalli – 620 027, Tamilnadu.
saranyasekar19@gmail.com

***Abstract* - In the present days, with the mount of IoT and a variety of smart devices such as sensors, mobile devices, satellites, etc., we are able to produce records at a volatile rate. Those devices may create records as an event and send it to some other devices for further processing. These takes place endlessly. These events may be processed at a time of the interval, resulting in a stream of events. This process is known as stream processing. In such circumstances, these events are generating at a very high speed at a rate of seconds or even milliseconds. So, we required to process these event streams at the same rate or higher processing rate. Big Data environment have been developed a number of techniques for processing streaming data such as Apache Storm, Apache Spark, Apache Kafka, Apache Flume, Apache Flink and so many. This paper makes a survey on above mentioned big data platforms for handling streaming data efficiently as well as effectively.**

***Keywords* – IoT, sensors, satellites, stream processing, Big Data Platforms, Apache Storm, Apache Spark, Apache Kafka, Apache Flume.**

I. INTRODUCTION

Today, the world is filled with innovative and emerging technologies which are used in our day to day life. These techniques have been developed for several purposes that depend on the domains. At present, most of the domains [1] are in need to produce enormous amount of data as its outcome due to the expansive growth in technology as well as population. Hence, the outcome is unpredictable at the moment. It may go extremely large in the subsequent days. It is the responsibility of the organization to construct an emerging technique to store and process such a vast amount of data. In order to tackle the issue, Big Data Platform offers various techniques for several purposes including storage of data in large amount, processing at high speed, supports structured, semi-structured and unstructured data [2] and missing values (if any). These salient features are the reason behind the usage of big data in many organizations.

This paper is organized with several sections, namely, section I gives a brief description about big data and its necessity in today's world. Section II represents an author's point of view on various techniques available for stream processing. Section III illustrates a diagrammatic representation of streaming techniques. Section IV offers some big data platforms which have been developed for processing streaming data. Section V concluded with some discussion gained from reviews.

II. LITERATURE REVIEW

In [1], a review has been made on big data analytics which represents the characteristics, tools and methods used to store and process such a large amount of data. This work targets on Hadoop and MapReduce, which are efficient in handling vast storage of data and also process streaming data in real-time. It supports scalability as MapReduce is a parallel programming model.

Different types of data and characteristics of big data have been given in [2]. The paper also made a comparative study on various technologies, namely, Pig, Hive and R. All the three have its unique features and supports large amount of data for

processing. Pig supports all the three types of data such as structured, semi-structured and unstructured data, whereas, Hive is mainly for large amount of storage and scalability and finally, R is used for statistical and graphical representation.

A survey has been done in [3], which describes various challenges, issues and tools in big data analysis. Data storage, scalability and security are the general challenges discussed in this work. Research issues such as handling streams of data generated from various IoT devices are more complicated in nature. Machine learning techniques have been developed to process such kind of streaming data. Another issue is scalability and demand of resources which could be offered by cloud computing through pay as you go method which leads to cost reduction. Apart from these challenges and research issues, this paper suggests some big data tools for stream processing, namely, Storm and Splunk as well as batch processing such as Apache Hadoop, MapReduce and more.

An overview of big data concepts and architectures of various technologies have been represented in [4]. The authors discussed about the several characteristics of big data such as volume, velocity, variety and so on. Due to the increased volume of data, the traditional database systems are not sufficient to store such a huge number of data. Hence, they suggest Hadoop framework for storage as it contains two components, namely, HDFS and MapReduce. Later, this paper is illustrated with architecture of big data integration ecosystem which consists of several layers such as data sources, data storage, data transformation, data processing, data analysis and data consumption are used to perform its unique functions.

The ref [5] is provided with some basic requirements of processing streaming data in real-time. Before processing the speedy data, it must requires the following properties: (i) the data must be keeps flowing continuously with respect to time (ii) handle streaming imperfections such as missing, delayed and out-of-order data as it is real-time in nature (iii) process streaming data immediately with very low latency in the range of milliseconds or even microseconds. Later, the paper is concluded with some software technologies for stream processing such as DBMS, Rule Engines and Stream Processing Engines with its merits and demerits.

Processing of streaming data using big data techniques have been quoted in [6]. Initially, the paper is organized with basic concepts of stream data generation and various conventional techniques such as clustering, classification and frequent items mining for processing. Also, they have mentioned about the gaps including vast amount of storage, as the size increases scalability leads to a major issue in those techniques. In order to fill those gaps, big data platforms such as Apache Storm, Spark Streaming, Apache Flume, Apache Kafka and so many have been developed for processing streaming data.

III. CATEGORIZATION OF STREAMING TOOLS IN BIG DATA

As mentioned earlier, the traditional techniques are not sufficient for storing as well as processing the streaming data. Naturally, it produces infinite (boundless) data which arrives at very high velocity [7] with timestamp. In this section, we categorize streaming tools available in big data for storing and processing is illustrated in Fig. 1.

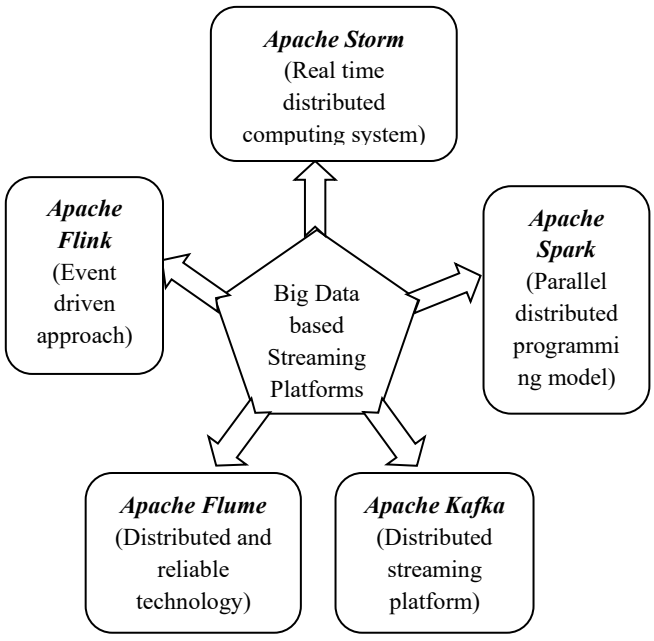

Fig. 1 Categorization of Big Data based Streaming Platforms

Big data streaming tools have been designed to handle those kind of speedy data in real-time. Moreover, as size of the data increases, scalability becomes a major issue. This could be solved with several big data platforms [8] which has been given in the following section.

IV. STREAMING TOOLS IN BIG DATA

Lots and lots of streaming tools have been developed for processing continuous flow of infinite data. The tools have been categorized under their unique functionalities [9] such as,

(i) Store enormous amount of data.
(ii) Processing at high speed data.
(iii) Can hold structured, semi-structured and unstructured data.
(iv) Handle lost and out-of-order data.
(v) Supports scalability in case of enlarge of size.

Hence, big data surroundings provide some efficient and innovative techniques which satisfy the above mentioned individualities. Some of these have been discussed in this section.

A) Apache Storm:

Apache Storm [10] is a free and open source distributed real time computation system. Apache Storm makes it easy to reliably process unbounded streams of data. It can be used in many applications, namely, real time analytics, online machine learning, continuous computation, distributed systems and more. Apache Storm is speedy as it supports *a million tuples processed per second per node.*

Storm has master-slave architecture [11]. There is a master server called Nimbus running on a single node called master node. There are slave services called supervisor that are running on each worker node. Supervisors start one or more worker processes called workers that run in parallel to process the input. Worker processes store the output to a file system or database. Storm uses Zookeeper for distributed process coordination. The overall architecture of Storm has been represented in Fig.2.

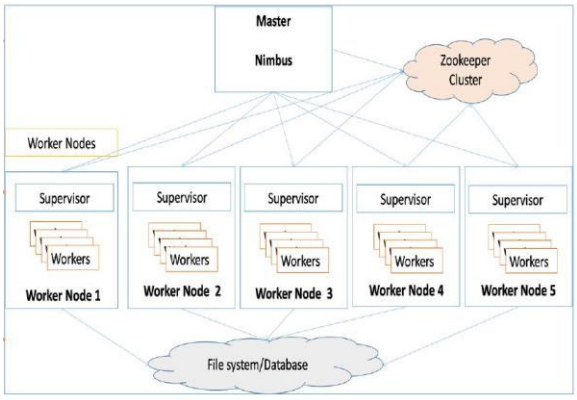

Fig. 2 Apache Storm Architecture

B) Apache Spark:

Apache Spark [12] has become one of the key big data distributed processing frameworks in the real world. Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on *Hadoop MapReduce* which includes interactive queries and stream processing. Spark uses Hadoop in two ways, one is storage and another one is processing. Since Spark has its own cluster management computation, it uses Hadoop for storage purpose only. The main feature of Spark is its *in-memory cluster computing* that increases the processing speed of an application. Spark is designed for batch applications, iterative algorithms, interactive queries and streaming.

Spark core API has four components, namely, Spark SQL, Spark Streaming, MLlib, and GraphX are illustrated in Fig. 3.

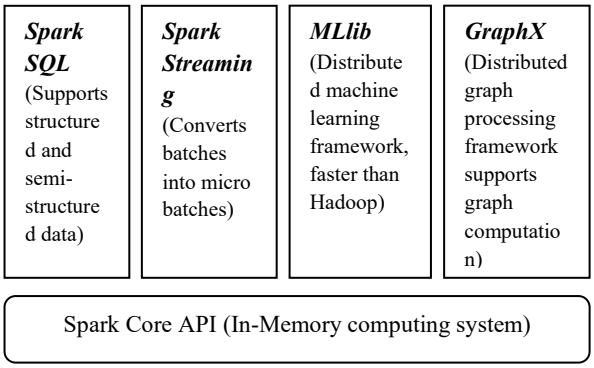

Fig. 3 Components of Spark API

C) Apache Kafka:

Apache Kafka is a streaming platform [13] which has three capacities listed here.

i) Publish and subscribe to streams of records.
ii) Store streams of records in a fault-tolerant manner.
iii) Process streams of records as they occur.

It has the ability to handle a large number of diverse consumers. Kafka is very fast, performs *2 million writes per sec*. Kafka is run as a cluster on one or more servers that can span multiple data centers. The Kafka cluster stores streams of *records* in categories called *topics*. Each record consists of a key, a value, and a timestamp. Kafka cluster has been represented with four components as follows: (i) Producer, (ii) Consumer, (iii) Stream Processors and (iv) Connectors are given in Fig. 4

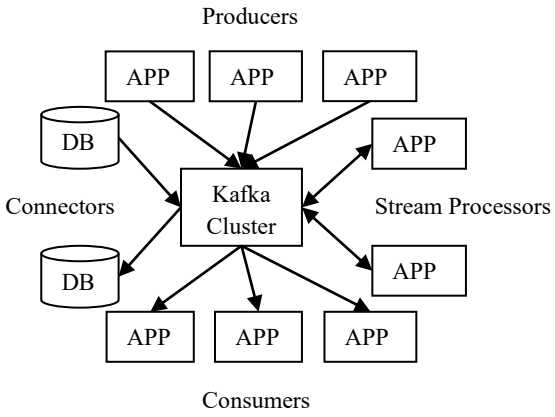

Fig. 4 Components of Kafka Cluster

Apart from these capabilities, Kafka is well suited for speedy data produced in many domains as it contains four streaming APIs [14] with it for processing the data in streaming model. Kafka Streaming APIs are listed in the following Fig. 5.

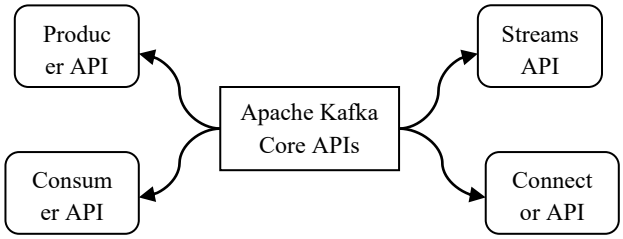

Fig. 5 Apache Kafka Core APIs

(i) Producer API: It allows an application to publish streams of records to one or more topics.
(ii) Consumer API: It allows an application subscribe one or more topics and then processing the streams of records.
(iii) Streams API: It allows an application to transform the input streams to the output streams.
(iv) Connector API: Executes reusable producer and consumer API with existing applications.

D) Apache Flume:

Apache Flume [13] is a tool or data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log data, events, etc., from various web servers to a centralized data store. It is a highly reliable, distributed, and configurable tool that is principally designed to transfer streaming data from various sources to HDFS.

A Flume agent [15] is a JVM process which has three components, namely, Flume Source, Flume Channel and Flume Sink through which events spread after initiated at an external source.

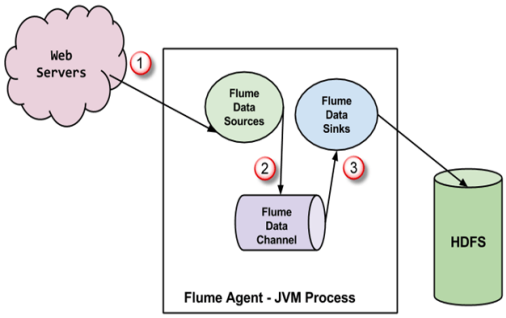

Fig. 6 Flume Architecture

E) Apache Flink:

Apache Flink [10] is a 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 [16], perform computations at in-memory speed and at any scale. Any kind of data is produced as a stream of events. Credit card transactions, sensor measurements, machine logs, or user interactions on a website or mobile application, all of these data are generated as a stream.

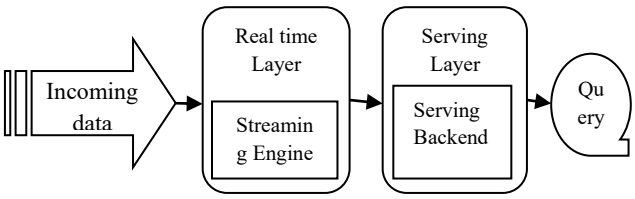

Fig. 6 Apache Flink Architecture

Apache Flink provides the capabilities to run real-time data processing pipelines in a fault-tolerant way at a scale of millions of events per second. Flink [17] is based on the Data Flow model i.e. processing the elements as and when they come rather than processing them in micro-batches. Dataflow allows flink to process millions of records per minutes at milliseconds of latencies on a single machine. Apache Flink provides exactly once processing like Kafka as well as support for event-time processing.

V. CONCLUSION

Due to the technological growth in adverse, the data generation leads to extremely large. Not only the size of the data, it is also depends on the data retrieval speed. Hence, this situation becomes complicated in conventional techniques. Moreover, scalability is not provided by ancient methods. This is the reason for move on to big data as it offers various streaming techniques for storing and processing large amount of data as well as scalability. This paper only focuses on streaming tools available in big data and provides some perception regarding its benefits. Hence, these tools turn out to be well-organized and promising in present days.

REFERENCES

1. Nada Elgendy and Ahmed Elragal, *"Big Data Analytics: A Literature Review Paper"*, Advances in Data Mining: Applications and Theoretical Aspects, Springer International Publishing, pp. 214-227, 2014.
2. Rachit Singhal, Mehak Jain and Shilpa Gupta, *"Comparative Analysis of Big Data Technologies"*, International Journal of Applied Engineering Research, Research India Publications, ISSN: 0973-4652, vol.13, pp.3822-3830, April 2018.
3. D. P. Acharjya and Kauser Ahmed P, *"A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools"*, International Journal of Advanced Computer Science and Applications (IJACSA), vol.7, No. 2, pp. 511-518, 2016.
4. P. Joseph Charles, S. Thulasi Bharathi and V. Susmitha, *"Big Data - Concepts, Analytics, Architectures - Overview"*, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, vol. 5, Issue. 2, pp. 125-129, February 2018.
5. Michael Stonebraker, Ugur Cetintemel and Stan Zdonik, *"The 8 Requirements of Real Time Stream Processing"*, ACM Publications, vol. 34, Issue. 4, December 2005.
6. Dmitry Namiot, "*On Big Data Stream Processing"*, International Journal of Open Information Technologies, Published by ResearchGate, ISSN: 2307-8162, vol.3, pp. 48-51, January 2015.
7. Albert Bifet, *"Real-Time Big Data Stream Analytics"*.
8. Bakshi Rohit Prasad and Sonali Agarwal, *"Stream Data Mining: Platforms, Algorithms, Performance Evaluators and Research Trends"*, International Journal of Database Theory and Application, ISSN: 2005-4270, vol. 9, Issue. 9, pp. 201-218, 2016.
9. Taiwo Kolajo, Olawande Daramola and Ayodele Adebiyi, *"Big data Stream Analysis: A Systematic Literature Review"*, Journal of Big Data, Published by Springer, 2019.
10. Vairaprakash Gurusamy and Subbu Kannan, *"The Real Time Big Data Processing Framework: Advantages and Limitations"*, International Journal of Computer Sciences and Engineering (IJCSE), E-ISSN: 2347-2693, vol. 5, Issue. 12, pp 305-312, December 2017.
11. https://www.simplilearn.com/introduction-to-storm-tutorial-video
12. Ounacer Soumaya, Talhaoui Mohamed Amine, Ardchir Soufiane, Daif Abderrahmane and Azouazi Mohamed, "*Real time Data Stream Processing Challenges and Perspectives"*, International Journal of Computer Issues (IJCSI), Online ISSN: 1694-0784, Print ISSN: 1694-0814, vol.14, Issue 5, pp.6-12, September 2017.
13. Z. Milosevic, Weisi Chen, Andrew Berry and Fethi A. Rabhi, *"Real-Time Analytics"*, 2016.
14. https://iteritory.com/beginners-guide-apache-kafka-basic-architecture-components-concepts/
15. https://data-flair.training/blogs/flume-architecture/
16. Nicoleta Tantalaki, Stavros Souravlas and Manos Roumeliotis, *"A review on Big Data real-time Stream processing and its scheduling techniques"*, International Journal of Parallel Emergent and Distributed Systems, Published by ResearchGate, March 2019.
17. Fatih Gurcan and Muhammet Berigel, "*Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks and Challenges"*, 2nd International Symposium on Multidisciplinary Studies and Innovatve Technologies (ISMSIT), Published by IEEE, Online ISBN: 978-1-5386-4184-2, 19-21 October 2018.