Emperor Journal of Applied Scientific Research

ISSN: 2581-964X **Mayas Publication®** **www.mayas.info**

Volume- VII **Issue- V** **May 2025**

A Structured Review and Comparative Study of Big Data Processing Frameworks: Hadoop, Spark, and Flink

C. Firza Afreen
Assistant Professor
PG Department of Computer Science
Islamiah Women's Arts and Science College
Vaniyambadi – 635 752, Tamil Nadu, India

Abstract

Big Data technology has become very important nowadays to deal with vast and complex datasets and to extract meaningful data. In order to manage data effectively, strong data processing frameworks are very important. This paper provides a Structured Review and Comparative Study of three famous big data processing frameworks, Apache Hadoop, Spark and Flink. The three selected frameworks have been analyzed in terms of its architecture, features, benefits, limitations and real-world use cases. Comparative analysis is conducted based on factors such as Processing Model, Performance and Latency, Fault Tolerance, Data Handling, Scalability, APIs, Ease of Use, Libraries, Community Adoption. The study highlights the strengths which is associated with each framework. Hadoop's batch-processing reliability, Spark's in-memory speed and Flink's true stream processing capabilities are discussed. The work done in this paper aims to guide the researchers and practitioners in selecting the best suitable frameworks according to their use-case requirements.

Keywords
Big Data Analytics, Apache Hadoop, Spark, Flink, Batch Processing, Stream Processing, Distributed Storage.

I. INTRODUCTION

In order to process very large amount of data, big data platform provides solution which includes lots of software, tools and hardware for the management of vast amount of data. Big Data platform solves all the requirements of different types of organizations despite the volume and size of data.Due to efficiency and effectiveness of Big Data platform lots of companies has nowadays using this technology in order to store organizations data. With the help of this technology the organizations will be able to convert the data present in different formats into structured form. Further, the organizations use the generated structured data in order to get the actionable business insights. Nowadays, both Open Source and Commercially available big data platforms are available.

Big Data Platform has different features. It supports different formats of data. The software will be able to deal with the data which is stored i.e with the data at-rest, and also with large amount of streaming data. This platform will be able to accommodate data at very fast speed. By using the tool provided, data will be cleaned, where the redundant and noisy data will be removed from the dataset. With the help of the tools provided, it is possible in this platform to convert data from one format to another format for getting insights from data, and also with the help of tools organizations will be able to analyze data accurately and efficiently.

Working of Big Data Platform

As its first step it deals with the process of data collection, where data will be collected from different sources like Sensors, Weblogs, Social Media and other databases.

The amount of data will be very large, here the second step takes place where in order to store the data, different storage spaces that is repositories will be used which is Hadoop Distributed File System, Google Cloud Storage or Amazon S3. Data Processing is the third step, where unwanted data will be filtered, noisy data will be cleaned, and data will be transformed from one format to another format, and also if required data aggregation will also takes pace. To carry out this third step data processing frameworks like Apache Hadoop, Spark and Flink will be used.

Once the data is processed, in fourth step Machine Learning Algorithms and Data Visualization Tools will be used in order to get the analytics from the processed data. The framework also ensures that the data is accurate and complete before using the data for analytics.

Overview of Frameworks

A. Apache Hadoop

For distributed storage and in order to do the batch processing of large amount of data Apache Hadoop Framework will be used, it is an open-source framework. Hadoop uses MapReduce Programming Model.

Architecture - HDFS (Hadoop Distributed File System)

HDFS Stores files in a decentralized manner, it stores files across multiple machines. Files with large size will be splitted into blocks of 128 MB and replicated by 3, default times. According to the need, users can customize the File block size and number of replication.

The HDFS uses MapReduce Programming Model.A programming model that processes data in two phases: In Map Phase input data will be processed and splitted into key-value pairs. The output of Map Phase is input to the Reduce phase, the Reduce phase processes intermediate data generated by Map phase.Another important component present in HDFS is YARN (Yet Another Resource Negotiator), it's main task is managing resources and job scheduling. This component makes sure, all the nodes have required resources for processing data and scheduler makes sure all nodes gets equal task.

The benefits of using this Framework are it is highly scalable, according to requirements number of devices and resources can be increased. As by default this framework creates 3 replication of our data, it is fault tolerant, even if one device fails, it is possible to retrieve data from other devices, while processing data if one node fails, automatically the other nodes will carry out the operation, as everything is automated here, and manual intervention is not required. Biggest benefit of using this framework is it will run on commodity hardware that is in low cost hardware devices. It has broad community support.

Few disadvantage too is associated in this Framework, First is Complex development, implementing Map-reduce is not a simple task, it needs careful analysis, if not done then accurate processing will not take place. This is one of the challenges which the developer will face. In some cases, more resources will be consumed to process the data.

Use Cases

In search engines it will be used, here the Hadoop indexes the web, and as a result of it users will get relevant result for their search, in very small amount of time. In Social Media it is used to analyze the user preferences and accordingly the recommendations will be given to user. In retail industries the transaction will be analyzed and the company will be able to predict the demand of the products in different places. In healthcare sectors it is used to store the patient records and predicts the disease, in banking sectors it is used for safer transactions and fraud detections, it is also used by Government for public data analysis.

B. Apache Spark

Spark is one of the analytical engines for processing big data, and it is mainly known for its in-memory computing. It gives high performance, and this analytical engine has the capacity to do both batch and stream processing.

Architecture Components

The Driver Program has been used in this Framework, its main task is to coordinate the Spark applications and task scheduling. Another component which is present in this framework is Cluster Manager, it manages the resources. Third important component which is present in this framework is Executors, as the name indicates, its main job is to run the tasks and return results to the driver. This framework also has RDD that is Resilient Distributed Datasets.

The benefit of using this Framework is mainly it does in-memory computing, also supports batch processing, stream processing and graph analytics. It also does parallel processing. For this framework Application Programming Interfaces are available in different languages like Java, Scala, Python and R. It is also possible to integrate this Spark Framework with other storage systems like HDFS, S3, HBase etc. For fault tolerance mechanism, instead of storing multiple copies on different machines, it uses lineage-based recovery mechanism.

The drawback which is associated with this framework is High-Memory Usage, because in this framework in-memory processing is taking place. Deployment and tuning of this framework is complex.

Use Cases

In factories, the Spark can be used in order to monitor thousand of sensors at the same time. When any Spike is detected in range, it sends an alert to shut down the machines. An airline system uses Spark, by using the information provided by Spark, the analyst will detect the reason for flight delay. The telecom company trains the machine first with the help of machine learning algorithms, trained data will be given to Spark, and then this framework will be able to categorize the customers. A Media company also uses Spark for processing video streaming logs, it collects the data, then cleaning of data will take place and then the cleaned data will be stored in the reporting system.A research lab also uses Spark for data sequencing to find markers of disease.

C. Apache Flink

Apache Flink is a free, powerful and open-source tool. It is mainly used to process data in real-time. Flink reacts to new data instantly. This framework is true stream processing, it process data as soon as it comes. Flink is greater framework for continuous events and quick reactions.

Architecture Components

First component present in this framework is Job Manager, as the name indicates it coordinates distributed execution. The task manager is another component which performs processing and maintains the local state of the system. Very important third component is Check pointing, it saves the state periodically, so that if failure occurs, recovery of processed tasks will take place from that point. It uses watermarks and it easily finds out, out-of-order data.

The benefit of using this Framework is it provides native support for the processing of complex events. It provides Checkpointing feature, it is true real-time stream processing, it is stateful, it provides low latency, it also handles late data and filters out out-of-order data efficiently. As other frameworks, it also provides fault tolerant options, it guarantees strong consistency.

The demerit of this Framework is the Programming model which is associated with this framework is complex, it has fewer libraries compared to other frameworks, it has lesser community support, compared to Apache Hadoop and Spark.

Use Cases

Flink reacts to risky transactions and with the help of this; it is easy to detect the fraudulent transactions easily. It updates the stock details instantly which is sold or returned, organizations will get idea on accurate inventory levels. This framework also adjusts the prices of the products live based on the factors like demand, competition, stock levels or user activity. It is also used to track information on how users do interaction with the website and based on this information, it personalizes the experience instantly. It is also used in Network Monitoring, it finds out the reasons for slow down of network, reasons of occurrence of error in network, it also finds out if any cyber attacks happen on network. It is used in Vehicle tracking for route optimization, emergency detection etc. It is also used for live Sports analytics for example, it streams real time scores.

Comparative Analysis

A. Processing Model

When it comes to Processing Model, in Hadoop Batch Processing will take place, it processes data in blocks using MapReduce Model, and it is also suitable for offline analytics. In Spark Batch and Micro Batch processing will take place, it processes data in-memory and it uses RDDs. In Flink True Stream processing will take place, it provides support for batch data.

B. Performance & Latency

Hadoop is slowest as compared to other two Frameworks, because it uses commodity hardware and it mainly depends on Disk, I/O heavily. Performance wise Spark is little fast or we can say its medium compared to Hadoop, because of in-memory processing, RDDs. Flink is much faster compared to Hadoop and Spark, reasons are, first it is event-driven and second reason is it offers true real-time analytics.

C. Fault Tolerance

All three frameworks supports fault tolerance mechanism, but the way fault tolerance mechanism has been implemented is different in these frameworks. In Hadoop by default 3 replication of files will be created and same will be stored at nodes in different locations. In Spark, by using RDD Lineage mechanism, lost data re computation will take place. In Flink for fault tolerance, Checkpoint mechanism has been implemented.

D. Data Handling

Hadoop supports Batch Processing, Spark supports In-Memory, Batch and Micro Stream processing, Flink supports In-Memory, Batch and Native-stream processing.

E. Scalability

As far as Framework Scalability Capability is concerned, in Hadoop it is possible to add nodes easily, Spark scales well with memory and Flink is scalable for real-time applications. In all three frameworks scalability is possible, but approach and methodology of scalability implementation is different.

F. APIs and Ease of Use

Hadoop APIs are available in Java mainly, Spark APIs are available in Scala, Java, Python and R and Flink APIs are available in Java, Scala and Python. As far as ease of development is concerned, Hadoop is complex, Spark is user-friendly, because it supports libraries for SQL, ML, Graph processing etc and Flink is slightly complex.

G. Libraries

Important Libraries supported by Hadoop Framework is Hive, Pig, HBase. Spark supports SQL, MLib, GraphX and Flink supports SQL, Stateful API. Spark has the most developed libraries compared to other Frameworks.

H. Community & Adoption

Popularity of Hadoop is, it’s on the way out, but still it is active due to its legacy users. The popularity of Spark is very high, because large community is using it and maximum industries available nowadays has adopted it. Flink is in growing state and it more popular among users who deal with real-time data.

Future Trends

If Real-Time Analytics is concerned, then Hadoop will provide limited support, Spark is improving in this area, and Flink is native and very strong already to deal with real time analytics. If Cloud-Native support is concerned, in Hadoop support is limited; in spark and Flink it is possible because of the availability of Kubernetes. Support of Machine Learning Integration is very weak in Hadoop, but the same is possible in Spark because of Built-in MLib Package, but in Flink in order to get the same, external library support is required.

II. CONCLUSION

The comparison of Hadoop, Spark and Flink shows that no single framework is universally greater, each framework deals with different needs of users based on the processing model and system architecture associated with frameworks. Each framework has its own architecture, merits and drawbacks. Hadoop is well-known for its batch processing and distributed storage done through HDFS. Spark framework supports batch and micro-batch processing, which makes it suitable to deal with applications like machine learning. Flink supports real-time stream processing; it offers low delay and high throughput. The analysis done here clearly shows that the choice of selecting and using Frameworks mainly depends heavily on demands of use-cases. This paper contributes a structured approach for selecting the accurate big data frameworks.

III. REFERENCES

[1] M. Parsian, Data Algorithms with Spark: Recipes and Design Patterns for Scaling Up. 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2022. ISBN: 978-1-4920-8238-5.

[2] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, Learning Spark: Lightning-Fast Big Data Analysis. 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2015. ISBN: 978-1-4493-5862-4.

[3] S. Ryza, U. Laserson, S. Owen, and J. Wills, Advanced Analytics with Spark: Patterns for Learning from Data at Scale. 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2017. ISBN: 978-1-4919-7294-6.

[4] F. Hueske and V. Kalavri, Stream Processing with Apache Flink. 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2019. ISBN: 978-1-4919-7429-2.

[5] M. Guller, Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis. 1st ed. New York, NY, USA: Apress, 2016. ISBN: 978-1-4842-0964-6.

[6] V. Ankam, Big Data Analytics: Real-Time Analytics Using Apache Spark and Hadoop. 1st ed. Birmingham, UK: Packt Publishing, 2016. ISBN: 978-1-78588-469-6.

[7]V. S. Agneeswaran, Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark, and More Hadoop Alternatives. 1st ed. Upper Saddle River, NJ, USA: Pearson Education, 2014. ISBN: 978-0133838251.

[8]N. Marz and J. Warren, Big Data: Principles and Best Practices of Scalable Real-Time Data Systems. 1st ed. New Delhi, India: Wiley India, 2015. ISBN: 978-9351198062.