AN EMPIRICAL COMPARISON OF BIG DATA FRAMEWORKS IN DATA RETRIEVAL

[1]Ajay Kumar, [2]Dr. Aman Kumar Sharma

[1]M.Tech. Student, [2]Professor

Department of Computer Science,

Himachal Pradesh University, Shimla, India

Abstract: With the advancement of technology, huge amount of real–time data has been generated referred to as big data. Cloud users produce large amount of data every day, big data has become one of the major research areas for handling such real time data. To process and analyse this vast amount of data, there many powerful frameworks like Hadoop Spark and Flink, which are mainly used in the context of big data. To find out, which framework is best according to processing and analysis is the main challenging task in Big data. This paper discusses comparison on Apache Hadoop, Apache Spark and Apache Flink.

Keywords: **Apache Spark, Big data, Hadoop, HDFS, MapReduce Apache Flink.**

I. INTRODUCTION

Organizations produces and storing large amount of data every day that is dynamic in nature, mainly in the web and online social networking applications such as Facebook, YouTube Twitter. To capture, searching, sharing, storing, analyzing and presenting the data. Big data was born. This is a concept for storing an enormous amount of data or information that is difficult to process with conventional database management frameworks. Big data is not just simple data, it is a set of technologies, frameworks and procedures that allow an organization to capture, processing and analyzing the data with no Extra cost. For data analysis and processing, Hadoop Spark and Flink is used [1].

II. LITERATURE REVIEW

Harshita Saluja, Pallavi Asthana et.al [2] This paper explores the five ways in which Big Data is characterized, methods of its classification, Big Data management and its role in effective implementation of e-governance like education, health care, revenue etc., where huge amount of data is generated. This data is useful in understanding the factors that can be monitored and analysed for improvement and betterment of current policies initiated by government. Big Data can provide tremendous opportunities in the area of e-governance by utilising it efficiently and in a secure way. This is an area where large data in stored and it needs to convert in to the implementation of the policies that are beneficial for the people of the country.

Sarala N R, Gagana R P, et.al [3] conducted survey of Big Data architectures, their handling techniques which handle a huge amount of data from different sources and improves overall performance of systems and also its applications which shows its importance and uses in the present IT world. It includes architecture using Hadoop HDFS, MapReduce distributed data processing over a cluster of commodity servers. Big Data is termed has any type of datasets which are so vast and compound which becomes difficult to process them using traditional data processing applications. While handling vast dataset different challenges may be faced by the user.

Diego García-Gil1, Sergio Ramírez et.al [4] is a comparative study for batch data processing of the scalability of two popular frameworks for processing and storing Big Data, Apache Spark and Apache Flink. Authors have tested these two frameworks using SVM and LR as learning algorithms, present in their respective ML libraries. This study has also implemented and tested a feature selection algorithm in both platforms. Apache Spark have shown to be the framework with better scalability and overall faster runtimes. Although the differences between Spark's MLlib and Spark ML are minimal, MLlib performs slightly better than Spark ML. Flink is a novel framework while Spark is becoming the reference tool in the Big Data environment. Spark has had several improvements in performance over the different releases, while Flink has just hit its first stable version. Although some of the Apache Spark improvements are already present by design in Apache Flink, Spark is much refined than Flink.

Alexandros Labrinidis, H.V. Jagdish [5] explored the controversies and debunk the myths surrounding Big Data. There have been many controversial statements about big data, such as Size is the only thing that matters. This study identify why big data is different from past Very large database techniques and what the most challenging aspects of big data are. Secondly, to determine how can industry and academia collaborate towards solving Big Data challenges. Finally, to consider the role of the data management community within the Big Data solutions ecosystem.

Mr. Piyush Bhardwaj, Abhishek Gupta, et.al [6] discussed about the Big Data, significance of Big Data, the problem of unstructured data which is generated in Big Data and 5 V"s problems of Big Data. Authors have done a comparative study of different tools on which unstructured data can be converted to structured data. The main objective of this comparison is not to criticize which is the best tool in Big Data, but to demonstrate its usage and to create alertness in various fields. Apache Hadoop is used to process the Big Data and other related projects of Hadoop. Map reduce programming model has been successfully used at Google for many different purposes.

III. APACHE HADOOP

Hadoop is an open source java framework for storing and running application on cluster of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. Hadoop ecosystem consists of services like HDFS (Hadoop Distributed System), MapReduce. HDFS inspired by Google File System (GFS) [7].

- **HDFS**

HDFS is a file system written in Java, stores big data, links data blocks logically, and streams data as high bandwidth to applications. HDFS separates file system metadata from application data. To achieve reliability in case of failure of nodes, HDFS replicates data across clusters. HDFS has master/slave architecture [7].

- **MapReduce**

Hadoop is believed to be reliable, scalable and fault-tolerant. MapReduce is good for applications that processing big data, but it performs bad in iterative algorithm and low-latency computations because MapReduce relies on persistent storage to provide fault-tolerance, and requires the entire data set to be loaded into system for running analytical queries. For that, Spark is introduced [7].

IV. APACHE SPARK

Spark is a cluster computing framework and an Engine for processing large scale data. It is alternative to MapReduce frameworks but Spark performs better by 10x in iterative machine learning tools and 20x faster for iterative applications. Spark is mainly used for processing real-time data or applied to iterative algorithms. Spark introduced RDDs (Resilient distributed databases) in memory and then performs variety of operations in parallel on these datasets [7].

V. APACHE FLINK

Apache Flink is an open-source system for processing the rea-time (streaming) and historical (batch) data. Flink is developed by apache software foundation. There are two API in Apache Flink i.e. the Dataset API for processing historic and the Datastream API for processing streamed data. Flink executes arbitrary dataflow in parallel ad pipelined manner. Flink cluster have three types of processes i.e. client, Job Manager and at least one task manager. Client takes the program code, convert it to a dataflow graph, and submit it to Job Manager conversion phase determines the data type of the data exchanged between operators and create serializers and other type specific code.

The Job Manager coordinates with the distributed execution of dataflow. It also tracks the state and progress of each operator and stream, schedule new operator, coordinates checkpoints and recovery. The actual data processing takes place in the Task manager. A TaskManager executes one or more operators that produce streams, and reports on their status to the JobManager [8].

VI. COMPARISON BETWEEN APACHE HADOOP APACHE SPARK AND APACHE FLINK

In table 1.1 depicts and highlighted the parameters of Hadoop, Spark and Flink. It compares the three frameworks.

Table 1.1 Comparison between Hadoop Spark and Flink [1] [2] [7] [8] [9] [10] [11]

Sr. No.	Parameters	Hadoop	Spark	Flink
1	**Distributed File System**	Own File System	HDFS, S3, Tachyon	HDFS, S3, Local FS
2	**Message Delivery guarantee**	Exactly-once	Exactly-once	Exactly-once
3	**Cost**	Less expensive	More Requirement of RAM increases cost	More Requirement of RAM increases cost
4	**Data Computation**	Disk-Based	In Memory	In Memory
5	**Hardware Requirement**	Commodity hardware	Mid to high-level hardware	Mid to high-level hardware
6	**Languages Supported**	Primarily Java, but other languages like C, C++, Ruby, Groovy, Perl, Python also supported using Hadoop	Java, Scala, python and R	Java and Scala
7	**Line of Code**	1,20000	20000	Line of code is lesser than hadoop

8	**Latency**	High	Medium	High
9	**Processing Model**	Batches	Micro-Batches	Single (one-by-one)
10	**Cluster Management**	YARN	YARN/Mesos/Standalone	YARN
11	**Framework**	Batch only processing	Hybrid (Batch and Stream processing)	Hybrid (Batch and Stream processing)
12	**Developers**	Apache Software Foundation	Apache Software Foundation, UC Berkeley AMP Lab	Apache Software Foundation
13	**Primarily Written in**	Java	Scala	Java
14	**Operating System**	Cross-platform	Microsoft Windows, OSX, Linux	Cross-platform
15	**Algorithm Used**	Word Count	Word Count	Word Count
16	**Available Since**	Dec 2011	Feb 2014	Dec 2014
17	**Cost**	Less Expensive Hardware	Requires a lot of RAM to run in-memory, gradually increases its cost	Requires a lot of RAM to run in-memory, gradually increases its cost
18	**Throughput**	Medium	High	Medium
19	**Fault Tolerance**	Yes (by replication)	Yes (Resilient Distributed Dataset)	Yes (Light weighted distributed snapshot/checkpoints)
20	**Windowing**	Doesn't support streaming so there is no need of window criteria	Time-based window criteria	Record-based, time-based or any custom user-declined Flink window criteria
21	**Users**	IBM, Amazon, LinkedIn, Cisco, Microsoft Google, VM ware, Facebook, HP, Adobe, Accenture etc.	Amazon, Yahoo!, NASA JPL, eBay Inc.	King – The crator of Candy Saga, ResearchGate
22	**Streaming Processing**	No	Spark Stream	Kappa Architecture
23	**Processing Speed**	Slower than Spark and Flink	100x Faster than Hadoop	Slower than Spark in Batch
24	**Programming Model**	MapReduce	Resilient Distributed Datasets (RDD)	Cyclic Dataflow
25	**Data Transfer**	Batch	Batch	Pipelined and Batch
26	**API**	Low Level	High Level	High Level

27	**SQL Support**	Hive, Impala	Spark SQL	Table API and SQL
28	**Graph Support**	No	GraphX	Gelly
29	**Machine Learning Support**	No	SparkML	FlinkML
30	**Duplication Elimination**	No	Processes every record exactly one time hence eliminates duplication	Processes every record exactly one time hence eliminates duplication

In table 1.1 it is presented that Spark is better in **processing**. This is in comparison to Hadoop and Flink as it is mentioned earlier that Hadoop and Flink processes slower, because Spark catches much of the input data on memory by RDD (Resilient Distributed Datasets) and keeps intermediate data in memory itself, eventually writes the data to disk upon completion or whenever required [11].
Spark is also better in throughput, latency and have less lines of code.
Throughput: The throughput is the number of events processed in this interval [11].
Latency: Latency is the time interval between simulation and response [11].
Line of Code: More number of lines produce more number of bugs and it will take much time to execute the program [11].
It is concluded that Spark is best framework as compare to Apache Hadoop and Apache Flink.

VII. CONCLUSION

In this paper, provide comparison of Hadoop, Spark and Flink on the basis of Literature, in terms of Books, Thesis Reports, Research Papers, Publications, Results generated from software tools, Publications report available either online or in printed form. The study concluded that Spark performs better than Hadoop and Flink. In times to come a better framework may be implemented with enhancement of better existing framework.

REFERENCES

[1] Sagiroglu Seref, Sinac Duyugu, "Big Data: A Review". In proc. Of International Conference on collaboration Technologies and System (CTS), 20-24 May, 2013, San Diego, CA, USA.

[2] H. Saluja, P. Asthana, S. Mishra, S. Kumar, and B. Hazela, 'Big Data In E-Governance Management'. No.12, pp. 321-325, 2018.

[3] S.N.R, 2Gagana R P, M. R, M. P. V, and R. L, 'Comprehensive Study on Big Data Analytics Sarala', Int. J. Comput. Sci. Eng., vol Eng., vol. 7, no. 15, pp.74-76, 2019.

[4] D. Garcia-Gil, S. Ramirez-Gallego, S. Gracia, and F. Herrera, 'A comparison on scalability for batch big data processing on Apache Spark and Apache Flink', Big Data Anal.m vol. 2, no. 1, pp. 1-11, 2017.

[5] A. Labrinidis and h. v. Jagadish, 'Challenges and Opportunities with Big Data', Proc. VLDB Endow., vol. 5, n0. 12, 2012.

[6] P. Bhardwaj, A. Gupta, M. Sharma, M. Gupta, and S. Singhal, 'A Survey on Comparative Analysis of Big Data Tools', Int. J. Comput. Sci. Mob. Comput., vol. 55, no. 5, pp. 789-793, 2016.

[7] Apache Hadoop

Available at: https://en.wikipedia.org

[8] Patel B Aditya, Birla Manashvi, Nair Ushma, "Adressing Big Data Problem Using Hadoop and MapReduce", in proc. Of NirmaUniversity International Conference on Enginering, 6-8 December, 2012, Ahmedabad, India [Online]

[9] Big Data

Available at: https://en,wikipedia.org/wiki/Big data

[10] Apache Hadoop

Available at: https://en.wikipedia.org

[11] https://data-flair.training/blogs/hadoop-vs-spark-vs-flink/