COMPARATIVE EVALUATION OF BIG DATA FRAMEWORKS ON BATCH PROCESSING

K.Anusha1, K.Usha Rani2

1 Research Scholar, 2 Professor

Department of Computer Science, Sri Padmavati Mahila University, Tirupati, India

1siri.bachina@gmail.com
2usharani.kurba@yahoo.com

Abstract: Big Data processing techniques analyze data sets at terabyte or even Petabyte scale. Offline batch data processing is usually having full power and full scale, dealing with random Business Intelligence use cases. The Map Reduce model is a framework for processing and generating large-scale datasets. Apache Spark is a fast and general engine for large-scale data processing over the Map Reduce model. The main feature of Spark is the in-memory computation. Recently a new framework called Apache Flink has emerged which focuses on distributed stream and batch data processing. Flink is built on an idea that many classes of data processing applications, including real-time analytics, batch data processing, and iterative algorithms can be expressed and executed as pipelined fault-tolerant data flows. Apache Flink is capable of high throughput and low latency processing of data. Hence, in this study an integration of Apache Flink and Cassandra is experimented to process batch (weather) data and compared with existing studies.

Keywords: Big Data, Batch Processing, Apache Spark, Apache Flink, Cassandra.

I. INTRODUCTION

In present days the growth of data has been increased tremendously. So there is a need of new frameworks which are capable to store and process these large amounts of data in an efficient manner. Weather data processing is one of the important aspects which need a better framework to store and process the data for effective prediction. Storing and processing large weather data sets in an efficient manner are very useful to predict the future weather conditions.

Batch processing has an extensive history within the big data world. Batch processing involves operating over a large, static dataset and returning the result at a later time when the computation is complete.
The datasets in batch processing are typically:

- **Bounded**: Datasets represent a finite collection of data
- **Persistent**: Data is almost always backed by some type of permanent storage
- **Large**: Batch operations are often the only option for processing very large sets of data.

Batch processing is well-suited for applications where complete set of records is required which involves more number of calculations [20].

Hadoop Map Reduce is the first distributed framework used for storing and processing large volumes of data emerging [22]. **Apache Hadoop** is a processing framework that exclusively provides batch processing [19]. In Hadoop, Map Reduce processes large data sets with a parallel and distributed algorithm. Map and Reduce are the two tasks that need to be performed in Map Reduce which requires a lot of time to perform these tasks thereby increasing latency [12]. In Map Reduce, data is distributed and processed over clusters with increased time and speed reduction. To overcome these problems in Hadoop, Spark and Flink came into existence to increase the processing speed.

Spark is an open source Apache framework which overcomes the issue of high latency in Hadoop. Spark is an in-memory processing framework in which no time is spent in moving the data in and out of the disk. Spark is faster than Hadoop Map Reduce as it improves processing speed with its in-memory computation [13].

Apache Spark is considered a replacement for the batch-oriented Hadoop system. But it includes a component called Apache Spark Streaming as well. The main difference between Apache Flink and Apache Spark is Flink was specifically built for streaming. Spark added Streaming onto their product later [15]. Apache Flink uses different APIs for batch data and streaming data. Hence, in this paper we experimented Flink for batch processing of weather data.

Apache Flink is an open-source platform for scalable batch and stream processing applications. Apache Flink is a framework for distributed, high-performing, always-available, and precise data streaming applications. Flink is faster than Apache Spark as it improves the overall performance by providing single run-time for streaming and batch processing applications [14]. Flink performs at large scale, running on thousands of nodes with very good throughput and latency characteristics.

Batch processing applications run efficiently as special cases of stream processing applications. The Flink Runtime process is shown in the following figure 1 [19].
Apache Cassandra was developed by Face book. Apache Cassandra, one of the NoSQL databases is a top level Apache project and is a distributed database for managing large amounts of structured data across many commodity servers. Apache Cassandra provides high availability with no single point of failure. Hence, in this paper we experimented processing of batch data by integrating Apache Flink with Cassandra.

In our previous study, Spark and Cassandra are considered for Batch Processing of weather data and observed that the results are better than Hadoop Map Reduce implementation [7]. In this study, Apache Flink and Cassandra integration is experimented and compared with our existing study.

The rest of the paper is organized as follows: Section 2 describes Literature Review; Section 3 represents the Methodology; Section 4 represents the Proposed System; in Section 5 experimental analysis are provided; and conclusions are provided in Section 6.

II. LITERATURE REVIEW

Many studies have been carried out on Big Data Techniques related to Weather Forecasting. Only few studies have been carried out on Batch Processing using Apache Flink. Those are:

Diego García-Gil et.al [1] provides a comparison on scalability for batch big data processing on Apache Spark and Apache Flink.


Sam Madden [5] explains that existing tools do not lend themselves to sophisticated data analysis at the scale many users would like.
Bhalchandra Bhutkar et.al [6] proposed that although traditional databases are useful for performing complex analytical queries in wide variety of applications they have several issues while handling the huge amount of data. NoSQL technologies like HBase, Cassandra, and Mongo DB have gained significance over the years because of their ability to handle big data in distributed environment. These technologies are mostly open source and provide a means of handling a significantly large volume of data with lower cost and easier management than traditional RDBMS.

K. Anusha et.al [22 ] provides an introduction about Big Data Characteristics and Hadoop Distributed File System.

III. METHODOLOGY

In this study, we experimented Apache Flink for processing weather data stored in Cassandra database. Brief description about Apache Flink and Cassandra are presented in this section.

Flink Implementation

Apache Flink is known as 4G of Big Data. Apache Flink is new generation Big Data processing engine which targets to unify different data loads. Flink is an alternative of Map Reduce and processes data 100 times faster than Map Reduce. Apache Flink is an open source platform and addresses various requirements efficiently.

Flink works in Master-Slave manner. Master acts like manager node of the cluster and slaves act like worker nodes. The client submits an application to the master which is the centrepiece of the cluster.

The master divides the work and submits to the slaves in the cluster. In this manner, Flink process the data at high speed and enhances the distributed computing power. Flink architecture is represented in following figure 2[17].
On master node, the master daemon of Flink “Job Manager” is configured and on all slave nodes the slave daemon of Flink “Node Manager” is configured. The Architecture is represented in the following figure 2(b) [17].
Flink provides two independent APIs. Dataset API for Finite data and DataStream API for Infinite data. These APIs sound very similar to Spark RDD and DStream in Spark. But they are not similar. The main difference is the APIs in Flink are optimized by using an optimizer [17]. The Flink Stack represented in the following figure 3 is taken from the study of Carbone et.al [2].

Flink guarantees that each message will be processed only once, so it is a good choice for implementing the Lambda architecture. Big Data is termed as processing large volumes of data of various forms which takes live streaming and historical data into account. One of the large computing clusters is to store and process these large amounts of data. This led to the development of new technologies of NoSQL. Hence, there is a need to integrate them in a standardized architecture like Lambda Architecture to enable Big Data. The Lambda Architecture is represented in the following figure 4 [21].
In this section, we present the main differences and similarities in the engines of both platforms in order to explain which are the best scenarios for one platform or the other.

Comparison between engines:

The first notable difference between both engines lies in the way each tool process streams of data.

Whereas Flink is a resident streaming processing framework that can work on batch data, Spark was formerly intended to work with stagnant data through its RDDs. Spark uses micro-batching to deal with streams. This method divides incoming data and processes little parts one at a time. The main benefit of this scheme is that the structure preferred by Spark, called DStream, is a simple queue of RDDs. This approach allows users to toggle between streaming and batch as both have the same API. However, micro-batching may not perform quickly enough in systems that require very low latency. However, Flink fits absolutely fine in those systems as it natively uses streams for any variety of workloads [1].
IV. PROPOSED SYSTEM

In this study, a method is experimented using both Apache Flink and Cassandra for processing of weather data stored in Cassandra database using Apache Flink and Cassandra Connector.

A. Flink-Cassandra Connector

In our previous research study, Spark-Cassandra integration was performed and the results are taken. In this study, Apache Flink is integrated with Cassandra for batch processing of Weather Data. Processing of data using Apache Flink is faster as Flink process data the same way, whether it is finite or infinite. But Spark was implemented for general purpose processing and is suitable for all big data applications. In case of Spark, it uses DStreams (Discretized Streams) for streaming data and RDD (Resilient Distributed Dataset) for batch data. Flink was intentionally developed as unified platform to process streaming and batch data. Flink uses Dataset API to process batch data.

Input Dataset: In this study, the weather data is collected from National Climatic Data Centre (NCDC) website. NCDC has data for every month with hourly basis [9].

Proposed Algorithm

1) Weather data collected from weather stations.
2) Run the Cassandra server.
3) Weather data is loaded into Cassandra.
4) Weather data is extracted from Cassandra using Dataset APIs in Flink.
5) Processing time is evaluated.

The algorithm is experimented with Flink Cassandra Connector on weather dataset.

V. EXPERIMENTAL ANALYSIS

By considering the proposed method (Flink-Cassandra Connector) the weather dataset is experimented and the result is tabulated in the following table 1.

TABLE 1

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Time Taken to import (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 MB</td>
<td>2</td>
</tr>
</tbody>
</table>
The current study is compared with our previous study where the same dataset is experimented on Spark and Cassandra and the results are tabulated in the following table for comparison.

**TABLE 2**
Comparison of Time (sec)

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Time taken to import(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hadoop Map Reduce</td>
</tr>
<tr>
<td>512 MB</td>
<td>30</td>
</tr>
</tbody>
</table>

The following figure 5 shows the graphical representation of analysis.

![Graphical representation](image)

**Fig 5: Spark Cassandra vs Flink Cassandra Benchmarking**

From the above tabulated results and graphical representation it is observed that Spark and Cassandra integration has better processing speed compared to proposed method. All architectures may not be suitable for all data sets. For all data sets it is not sure that Apache
Flink is better than Spark [1]. Hence it is observed that for this weather data set Spark yields better results than Apache Flink.

VI. CONCLUSIONS

So many frameworks are available for Big Data Analytics. All architectures may not be suitable for all data sets. In this study, a method is proposed for storage and processing of weather data using Big Data techniques. The method is experimented on weather data set using an integration of Apache Flink and Cassandra and the result is compared with both Spark-Cassandra Connector and Hadoop Map Reduce implementations. Apache Spark has shown to be the framework with overall faster runtimes. For the experiments, the same dataset is being used. Experiment results shown that processing of weather data stored in Cassandra using Spark has reduced processing time than Flink. Hence, it could be concluded that Spark-Cassandra Connector process the finite size weather data stored in Cassandra database with reduced time compared to Flink-Cassandra Connector.

REFERENCES

18. https://www.slideshare.net/datamantra/introduction-to-apache-flink-56892153