BIG DATA TECHNOLOGIES
Assignment 8

1) There were two main issues with Twitter's ETL process that affected data analytics:

1. It was difficult to design and maintain ETL pipelines.
2. Because business intelligence was performed on data that was only a day old, ETL pipelines created complexity. This was caused by the nightly running of ETL operations, which is standard routine. As business activities advanced, organizations required ever-fresher data to make faster conclusions.

Problems with ETL operations can have a substantial influence on data analytics. For example, incorrect or inadequate data can lead to misleading conclusions. Storage of data can make it difficult to have a comprehensive understanding of the business. Furthermore, complex or unstable ETL processes might make it difficult to deliver timely and effective insights.

Twitter's ETL process was unable to keep up with business demands. The pipelines were difficult to track, and the data was too old to make real-time decisions.

2) Twitter is a good example of a lambda architecture application for counting tweet views. This is because we want historical counts as well as real-time data on tweet views.Twitter generates billions of tweets every day, and in order to power its features and services, it must be able to manage and analyze this data in real time. Tweet data, for example, is used by Twitter to produce trending topics, recommend accounts to users, and encourage its search feature.

For example, even if a Donald Trump tweet is currently generating a lot of attention, we would be interested about how many views it has received over time.Twitter would be able to process and analyze twitter data in real time thanks to the lambda design.

3) The lambda architecture has two limitations:

a. There was a delay in the logging pipeline: Twitter's effect information is now being logged in interface logs. For aggregation, the logs are loaded into a Hadoop data warehouse via a multi-phase pipeline. Even under ideal conditions, the Map Reducing batch processing layer—where these types of aggregations were already common—caused pipeline logging delays, resulting in logs that were a couple of hours late.

b. Complexity and optimization: Because of the lambda architecture, everything must be generated twice on a normal basis and once for the on-demand platform. In many cases, the approaches used are vastly different. It is simple to determine the correlation of a set (e.g., the total number of tweets in a particular hour) in batching, and this quantity can be used for further computations and optimisations.

The lambda architecture is an effective device for processing and analyzing enormous amounts of data in real time. However, it is an advanced and costly approach. Twitter discovered that the lambda architecture was not well-suited for all of its data processing needs, and it has now abandoned the lambda design instead of a hybrid approach.

4) Kappa architecture is a data processing architecture that was created for making real-time and batch data processing easier.

The Kappa architecture is a software architecture based on activities that can manage any form of data at any size and in real time, including jobs that require both analysis and transactions. The Kappa architecture is a streaming-first infrastructure implementation style that processes data from batch, streaming, near-real-time, and Internet of Things sources (such as change data collection) using Apache Kafka communication technologies. Analytics can be performed in real-time by taking data from the messaging system, enhancing it, and then publishing the modified data back to the messaging system using a stream processing engine (such as Apache Spark, Apache Flink, or another one). Data is processed in order to be used for self-service analytics and machine learning (ML), tracking, visualizing, prediction and preventative maintenance, and alerting applications.

5) Apache Beam is an open-source, unified technique for defining pipelines for distributed processing of streaming and group data. Apache Beam's programming design simplifies large-scale data processing principles. Users can construct an application that describes the pipeline by using one of the Apache Beam SDKs.

Managing a team of employees, dividing up datasets, and other comparable activities are among the more complex aspects of distributed processing that the Apache Beam architecture protects you from. These little aspects are entirely under Dataflow's control.

Characteristics that identify Apache Beam:

a.Unified: Use a single coding technique for batch and streaming use cases.

b.Portable: Pipelines can be used in a variety of scenarios. Different accelerators are used in various processing situations.

c.Adaptable: creating separate IO interfaces, transformation modules, and SDKs.

Submitted by:
Sailavanya Narthu
A20516764
snarthu@hawk.iit.edu