Chaitanya Durgesh Nynavarapu
A20561894

ASSIGNMENT-8

BIG DATA (CSP 554)

EXERCISE 1:

1. Extract-transform-load (ETL) takes transactional business data (think of data collected about the purchases you make at a grocery store) and converts that data into a format more appropriate for reporting or analytic exploration. What problems was encountering with the ETL process at Twitter (and more generally) that impacted data analytics?

- ➢ Problems with ETL at Twitter: Twitter encountered issues with ETL processes primarily due to latency and maintenance challenges. ETL pipelines were difficult to build and maintain, and they introduced significant delays, as data was often processed in nightly batches, meaning that business intelligence was based on outdated data. Increasing the frequency of ETL jobs only stressed these pipelines further, sometimes beyond their capacity.

2. What example is mentioned about Twitter of a case where the lambda architecture would be appropriate?

- ➢ Appropriate Lambda Architecture Use Case at Twitter: An example where the lambda architecture would be appropriate at Twitter is in counting tweet impressions. This requires both real-time updates as users interact with tweets and historical counts for comprehensive analytics.

3. What did Twitter find were the two limitations of using the lambda architecture?

- ➢ Limitations of the Lambda Architecture at Twitter: Twitter found two main limitations with the lambda architecture: complexity and maintenance overhead. The architecture required maintaining two separate codebases for batch and real-time processing, which increased complexity and made it difficult to ensure consistency between the two systems.

4. What is the Kappa architecture?

- ➢ Kappa Architecture: The Kappa architecture is a single-layer data processing architecture designed to handle all data processing as streams. Unlike lambda architecture, it eliminates the batch layer, allowing all processing to be done in real-time using a stream processing engine. This approach simplifies architecture by using a single technology stack for both real-time and historical data processing.

5. Apache Beam is one framework that implements a kappa architecture. What is one of the distinguishing features of Apache Beam?

- ➢ Distinguishing Feature of Apache Beam: One distinguishing feature of Apache Beam is its ability to handle both bounded (batch) and unbounded (streaming) datasets within a unified programming model. It provides a rich API that distinguishes between event time and processing time, allowing for sophisticated handling of data streams with varying arrival times.

EXERCISE 2:

"Real-time Car Tracking System Based on Surveillance Videos" by Seungwon Jung, Yongsung Kim, and Eenjun Hwang explores a sophisticated framework designed to enhance vehicle tracking using diverse video surveillance devices such as CCTV, dashboard cameras, and drones. The authors address the limitations of traditional single-source video analysis by proposing a comprehensive system that integrates data from multiple sources to improve accuracy and efficiency in vehicle tracking.

Introduction and Motivation

The study begins by highlighting the widespread use of video surveillance technologies for security and monitoring purposes. Traditional methods, which analyze frames from individual video sources, are limited due to their restricted coverage. The authors propose a real-time car tracking system that combines data from various video sources, thereby providing a more holistic tracking solution.

System Architecture

The proposed system, named Integrated Video-based Automobile Tracking System (IVATS), consists of three main components: Frame Distributor (FD), Feature Extractor (FE), and Information Manager (IM). The FD distributes video frames from various devices to processing nodes using Apache Kafka, ensuring efficient data transfer. The FE extracts key vehicle features such as license plate numbers, location, and time from the frames. The IM stores these features in a database built on HBase, handling user queries by retrieving relevant information efficiently.

Methodology

The system is built on a distributed processing framework for scalability and fault tolerance. Apache Kafka is used by the FD to manage the high volume of data generated by real-time video feeds. Image processing techniques are employed by the FE to extract vehicle features, transforming frames into grayscale images and applying filters to identify regions of interest. License plate recognition is performed using optical character recognition (OCR) technology. The IM utilizes an index structure combining R-trees and Hilbert space-filling curves for efficient spatial data retrieval.

Experiments and Results

Experiments conducted to evaluate the system's performance demonstrated its ability to handle large volumes of data efficiently and provide accurate vehicle tracking information. Visualization tools integrated into the system allow users to view vehicle trajectories on maps, enhancing usability.

Discussion and Conclusion

The study concludes that IVATS offers significant improvements over traditional car tracking methods by integrating multiple video sources and employing advanced data processing techniques. While the system provides real-time processing capabilities adaptable to various surveillance applications, its implementation requires substantial storage and processing capabilities due to the large volume of data involved.

Overall, this research contributes to the field of video-based surveillance by providing a scalable solution for real-time vehicle tracking, highlighting the potential for further advancements in integrating diverse data sources for enhanced security applications.