International Journal of Computer Applications (0975 – 8887)
Volume 185 – No. 9, May 2023

Kafka-based Architecture in Building Data Lakes for Real-time Data Streams

Kiran Peddireddy
Senior System Engineer
Cox Automotive
Atlanta, USA

ABSTRACT

The purpose of this paper is to investigate how Kafka can be used to construct data lakes for real-time data processing. Kafka has gained widespread popularity as a data ingestion and processing tool that offers scalability, fault tolerance, and flexibility. The benefits of utilizing Kafka in a data lake architecture are analyzed, as well as the procedures involved in utilizing Kafka in a data lake architecture. In addition, a case study is provided of a major financial institution that utilized Kafka to establish a data lake. The significance of Kafka in modern data processing is emphasized in this paper, as well as its worth in developing data lakes for real-time data processing.

General Terms

Data engineering, Big data, Apache Kafka, Data Lake.

Keywords

Kafka, KSQL, Data Lake.

1. INTRODUCTION

In contemporary data architectures, data lakes have emerged as a crucial element that allows organizations to store and analyze large amounts of data from diverse sources. Kafka is a distributed streaming platform that can be utilized as a tool for data ingestion and processing in a data lake architecture. The aim of this paper is to investigate the advantages of incorporating Kafka in a data lake architecture and to explain how it can be implemented. Kafka's strengths as a tool for data ingestion and processing, its scalability, and its fault-tolerant features make it a popular choice for data lake implementation. The paper examines how Kafka can be used to handle real-time data streams, which is particularly useful in data-driven applications such as fraud detection, real-time risk analysis, and customer behavior analysis. By analyzing the integration of Kafka in data lake architecture, this paper contributes to the growing body of knowledge on modern data processing technologies [2].

2. OVERVIEW OF KAFKA

Kafka is a distributed streaming platform that is open-source and capable of handling massive amounts of data from multiple sources. It is composed of several brokers that collaborate to store and process data. Kafka has high throughput and low latency capabilities, and its horizontal scaling capacity makes it perfect for data ingestion and processing in a data lake architecture [2].

This platform has become increasingly popular for real-time data processing and data streaming in modern data architectures. It can handle multiple types of data, including structured, semi-structured, and unstructured data, making it more versatile and flexible for use in diverse business environments. Kafka's design ensures high availability and fault tolerance, ensuring that data is not lost in the event of a failure. Its scalability allows it to accommodate increasing data volumes and processing requirements [2].

3. BENEFITS OF USING KAFKA IN A DATA LAKE

Kafka offers several advantages when used in a data lake architecture, such as:

Real-time data processing: Kafka's ability to ingest and process data streams in real-time enables organizations to react swiftly to changing situations.

Scalability: Kafka's horizontal scaling capability allows it to add more brokers to accommodate growing data volumes or processing demands.

Fault tolerance: Kafka's high availability and fault-tolerant design guarantees that data is not lost in the event of a failure.

Versatility: Kafka can handle multiple types of data, including structured, semi-structured, and unstructured data. This flexibility makes it easier to integrate Kafka into existing data architectures and to expand the range of data types that can be processed and analyzed.

In summary, integrating Kafka into a data lake architecture provides significant benefits, such as real-time data processing, scalability, fault tolerance, and versatility, that enable organizations to handle large volumes of diverse data types and respond quickly to changing business requirements.

4. USING KAFKA IN A DATA LAKE SETUP

Place Kafka can be used in a data lake architecture through the following steps:

Setting up Kafka: This entails setting up a Kafka cluster with multiple brokers, which can be deployed either on-premises or in the cloud using services such as Amazon MSK or Confluent Cloud.

Creating Kafka topics: These are used to organize data in Kafka based on the type of data or the source system.

Ingesting data: Various methods can be used to ingest data into Kafka, including Kafka producers or Kafka Connect. Kafka Connect can be configured to ingest data from different sources such as messaging systems, databases, and file systems.

Processing data: Kafka can process data in real-time by using Kafka Streams, which is a Java library that allows for real-time data processing.

Storing data: Kafka can store data temporarily or integrate with a data storage system like Hadoop or Amazon S3.

Implementing Kafka in a data lake architecture can enable organizations to achieve real-time data processing, scalability, fault tolerance, and flexibility. It can also simplify the process of migrating from legacy data warehouses to modern data lakes while ensuring factors such as security, data governance, and data quality are given due consideration.

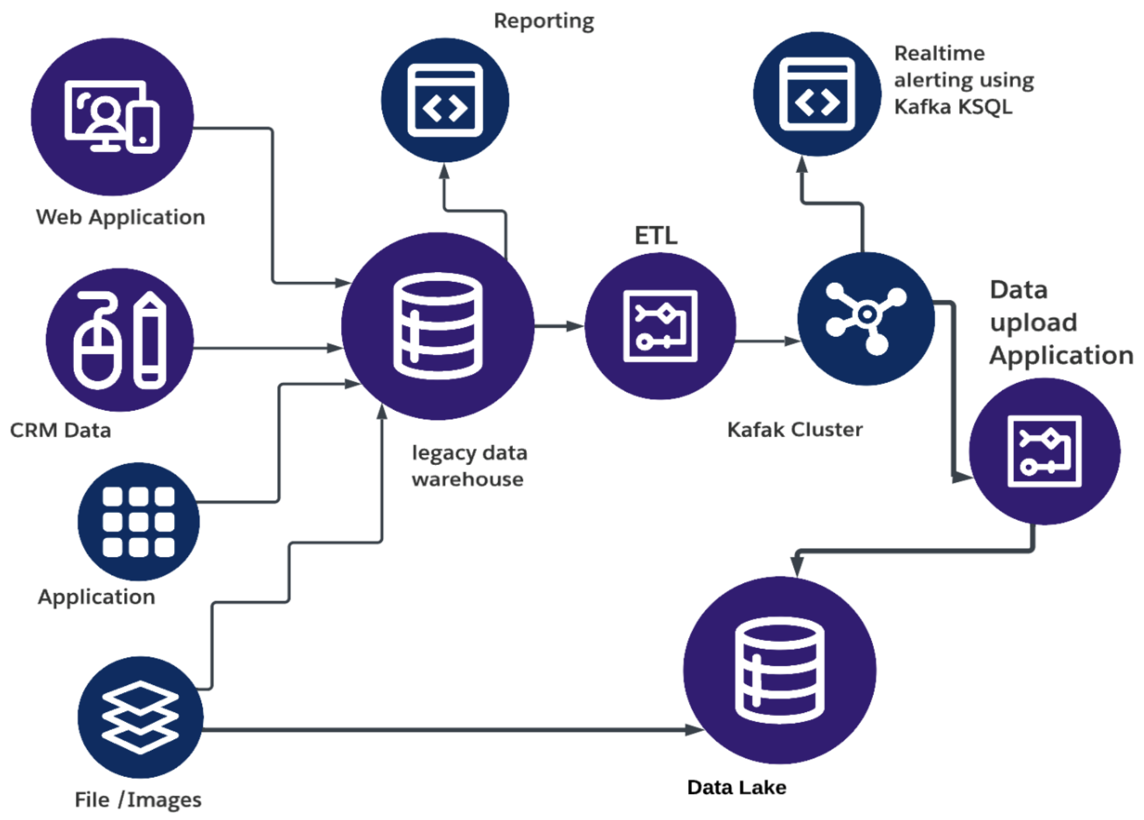

Fig 1: Kafka Data Stream for migration Data Lakes

5. MIGRATION FROM LEGACY DATA WAREHOUSES TO MODERN DATA LAKE

The traditional data warehouses were once used as a standard approach for storing and processing data, however, modern data lakes have become popular due to the increased demand for real-time data processing and big data. Migrating from legacy data warehouses to modern data lakes can be a challenging task, but by using Kafka, the process can be simplified.

The first step in this migration process is to identify the data that needs to be migrated, including data sources, models, and quality. The data can be ingested into Kafka using Kafka Connect. With Kafka, the data can be processed in real-time, enabling organizations to quickly respond to changing conditions. Complex data transformations and analysis can be performed using Kafka Streams. After processing, the data can be stored in a data lake that is built on Hadoop or another distributed file system [8] [9].

During migration, it is important to ensure data security, governance, and quality. Best practices such as encryption, authentication, and access controls should be employed to secure Kafka. Data governance should be implemented to properly manage, monitor, and audit data. Data validation, cleansing, and enrichment techniques can be used to ensure data quality.

6. REAL-TIME DATA PROCESSING WITH KAFKA

Kafka is a valuable tool for real-time data processing in a data lake architecture. With its ability to ingest data streams in real-time, process the data using Kafka Streams or other streaming frameworks, and store the processed data in a data lake, Kafka enables organizations to respond quickly to changing conditions or events. A case study of a large financial institution highlights the use of Kafka in building a data lake for storing and processing large volumes of data from various sources. The institution set up a Kafka cluster on-premises with multiple brokers for fault tolerance and scalability. Kafka topics were created based on data type and source system [6]. Data was ingested using Kafka Connect, configured to ingest data from databases, messaging systems, and file systems. Kafka Streams was then used to process the data in real-time, and the processed data was stored in a data lake built on Hadoop. The data lake was used for various tasks, including risk analysis, fraud detection, and customer behavior analysis. Kafka's scalability, fault tolerance, and real-time processing capabilities make it an ideal tool for building data lakes in a modern data architecture [5][7].

7. CONCLUSION

Kafka is a powerful tool that offers various capabilities for data ingestion, processing, and storage in a data lake architecture. Kafka's scalability, fault tolerance, and real-time processing features make it an excellent choice for modern data architectures. Additionally, Kafka simplifies the process of migrating legacy data warehouses to modern data lakes. By utilizing Kafka for data ingestion, processing, and storage, organizations can construct a more adaptable, flexible, and cost-effective data architecture. To ensure the successful integration of Kafka into a data lake architecture, organizations must consider factors such as security, data governance, and data quality. Kafka should be secured using best practices, such as encryption, authentication, and access controls. Data governance measures should be implemented to guarantee that data is appropriately managed, monitored, and audited. Data quality should also be maintained by applying data validation, cleansing, and enrichment techniques.

In conclusion, Kafka is a crucial component of modern data architecture, and its incorporation into data lakes can offer several benefits, such as real-time data processing, scalability, and fault tolerance. By leveraging Kafka's capabilities, organizations can construct more efficient, cost-effective, and scalable data architectures that can meet the demands of today's data-driven business environment.

8. REFERENCES

[1] Kiran Peddireddy. (2023). Book Title: “Enterprise Data Integration and Streaming Using Kafka, ActiveMQ, and AWS Kinesis”- ISBN -13 979-8372725218.

[2] Apache Kafka Documentation. (2021). Retrieved from

https://kafka.apache.org/documentation/

[3] Yu, T., Li, Y., Li, X., & Zhang, J. (2019). A Real-Time Customer Complaint Management System Based on Big Data Analytics. Journal of Computational Science, 31, 15-24.

[4] H. Wu, Z. Shang, G. Peng and K. Wolter, "A Reactive Batching Strategy of Apache Kafka for Reliable Stream Processing in Real-time", 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), pp. 207-217, 2020.

[5] K. Peddireddy and D. Banga, "Enhancing Customer Experience through Kafka Data Steams for Driven Machine Learning for Complaint Management," International Journal of Computer Trends and Technology, vol. 71, no. 3, pp. 7-13, 2023, doi: 10.14445/22312803/IJCTT- V71I3P102.

[6] G. van Dongen and D. V. D. Poel, "A Performance Analysis of Fault Recovery in Stream Processing Frameworks", IEEE Access, vol. 9, pp. 93745-93763, 2021.

[7] J. Kreps, N. Narkhede, J. Rao et al., "Kafka: A distributed messaging system for log processing", Proceedings of the NetDB, pp. 1-7, 2011.

[8] H. Mehmood et al., "Implementing Big Data Lake for Heterogeneous Data Sources," 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), Macao, China, 2019, pp. 37-44, doi: 10.1109/ICDEW.2019.00-37.

[9] J. C. Couto and D. D. Ruiz, "An overview about data integration in data lakes," 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 2022, pp. 1-7, doi: 10.23919/CISTI54924.2022.9820576.