**Title : Real Time Smart Bank Data Streaming Capture**

**GROUP - 03**

Sindhu Nagesha (017419987)

Prayag Nikul Purani (017416737)

Syed Faraaz Ahmed (017428619)

Sai Vivek Chunduri (017435301)

Department of Applied Data Science,

San Jose State University

DATA 228: Big Data Tech and Applications

Submitted to: Vishnu S Pendyala

March 08, 2024

### **Abstract :**

In the era of digital banking, real-time data processing has become imperative for delivering personalized and responsive services to customers. This project delves into the realm of real-time data analytics for the banking sector, focusing on the implementation of a smart bank data pipeline. Employing Change Data Capture (CDC) techniques, the project captures and analyzes real-time data alterations originating from diverse banking systems. A robust technological stack comprising Docker, Postgres, Debezium, Kafka, Apache Spark, and Slack is utilized to ensure efficient data streaming and analysis. Apache Spark processes the streamed data in real-time, facilitating swift analysis and the generation of actionable insights. Integration with Slack enables instant notifications and alerts, enabling banking personnel to promptly address critical events such as fraud detection or transaction anomalies.Through the implementation of this intelligent bank data pipeline, organizations can glean valuable insights into customer behavior, swiftly identify fraudulent activities in real-time, and deliver tailored banking experiences. Ultimately, this project underscores the transformative potential of real-time data analytics in revolutionizing the banking sector and elevating customer satisfaction levels.

### **Motivation for the project:**

The motivation for implementing a real time streaming project are:

Real-time Data Processing: In this fast-paced modern world, businesses require the ability to make decisions based on the most up-to-date real-time data. This project enables the capture of data changes as they occur, facilitating real-time data processing.

Data Replication: Capturing the changes in real time can be an essential component of data replication strategies. It ensures that modifications made to one database are automatically propagated to another, maintaining synchronization between databases.

Audit Logging: CDC can be utilized to maintain an audit trail of all changes made to a database. This capability is valuable for debugging, analyzing user behavior, and preserving a historical record of data transformations.

Data Integration: This project can be used to integrate data from diverse sources into a centralized data warehouse or data lake in real-time.

### **Brief literature survey :**

1. **Apache Spark: A Big Data Processing Engine**

<https://www.researchgate.net/publication/339176824_Apache_Spark_A_Big_Data_Processing_Engine>

In this paper we learn how Big data refers to an excessively large amount of datasets that are used to computationally reveal patterns and trends. In order to analyze and ﬁnd knowledge from this bulk of data, a processing framework is required. There are various types of commonly used big data frameworks such as Apache Hadoop, Apache Storm, Apache Spark, Apache Flink etc. In this paper we learn about Apache Spark’s batch processing and stream processing abilities, use cases, ecosystem, architecture, multi-threading and concurrency capabilities and lastly the use of Spark in emerging technologies.

1. **Information Security in Big Data: Privacy and Data Mining**

<https://ieeexplore.ieee.org/abstract/document/6919256>

Data mining can extract valuable knowledge and patterns from large datasets, but also raises privacy concerns about sensitive personal information being disclosed. Privacy Preserving Data Mining (PPDM) is a research area focused on modifying data to enable effective data mining while protecting sensitive information. Most PPDM work has looked at reducing privacy risks in the data mining operations phase. However, privacy threats can arise in other phases like data collection, publishing, and delivery of mining results. The paper takes a broader perspective, identifying four types of users involved in data mining applications: data provider, data collector, data miner, and decision maker.

For each user type, the paper discusses their privacy concerns and methods to protect sensitive information throughout the knowledge discovery process. In addition to reviewing privacy-preserving approaches per user role, the paper also covers game theoretical approaches that analyze interactions and valuations of sensitive information among different users. The goal is to provide insights into PPDM by differentiating the privacy responsibilities of different user roles in safeguarding sensitive information throughout the data mining pipeline.

1. **Beyond Batch Processing: Towards Real-Time and Streaming Big Data**

<https://www.mdpi.com/2073-431X/3/4/117>

This paper examines the limitations of traditional batch processing systems like Hadoop MapReduce in handling real-time queries, interactive jobs, and continuous data streams. It reviews emerging solutions for real-time processing, such as in-memory computing platforms and real-time query engines, as well as dedicated stream processing frameworks like Storm and S4. Through experimental results, the paper demonstrates the performance advantages of these new solutions over Hadoop for real-time and streaming workloads. The author concludes that while batch processing with Hadoop is mature, in-memory computing approaches like Spark are becoming essential to meet the growing real-time and streaming needs of big data applications.

1. **Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges**

<https://ieeexplore-ieee-org.libaccess.sjlibrary.org/document/8567061>

This paper outlines the importance of real-time processing in today's technology landscape, especially in big data applications. It introduces a lifecycle for real-time big data processing, covering phases like data ingestion, storage, stream processing, analytical data store, and analysis/reporting. The document explores various tools such as Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, Hbase, Hive, Cassandra, Splunk, and Sap Hana, associating them with different lifecycle stages. Additionally, it addresses challenges like handling large and diverse data, ensuring consistency, scalability, real-time processing, data visualization, skill requirements, and privacy/security. The paper aims to provide insights into the lifecycle, tools, and challenges of real-time big data processing.

1. **KAFKA:The modern platform for data management and analysis in big data domain**

<https://ieeexplore.ieee.org/abstract/document/8343593?casa_token=-2a8CZ48gFcAAAAA:zP12qnjnssH7pT01K1Xsu6vHXYKt2qK_e6L7RytfVB3Wr9QGW55AZ5b9GnigeTqixZEeywe39gw>

The paper talks about how dealing with real-time data nowadays is quite complicated, with many technologies that need to work together. It suggests using Apache Kafka, which is like a smart system for handling streams of data. It acts as a messaging system and is good at storing data reliably. Kafka is useful for two main things: moving data between systems in real-time and creating applications that work with live data streams. It works on multiple servers, storing data in categories called topics, each having a key, a value, and a timestamp. The paper explains Kafka's structure and gives examples of how it helps solve problems in the Big Data era by using streaming solutions.

**Methodology :**

### 

Architecture of how our project works, and we will be implementing the tools in order to reach our goal.

### **Software Requirements :**

Environment → docker

Connector → Debezium

Storage → PostgreSQL

Reading → Kafka

Processing → Apache Spark, Apache Storm, ksqlDB, Flink

Visualization → Apache Superset/elasticsearch/PowerBI/Tableau

### **Deliverables and milestones :**

1. The system architecture is the main part and the implementation of the the tools will helps us to learn different corners of the courses and apart from this the smart bank will be integrated with the the real-time data and we are also, planning to integrate the data from the sensors from the end users so we can process the data from this part and can tell whether something is fishy or not so, the problem told in the class could be overcome.
2. So, the data set taken into account will be the application data, the transaction data and the data from the user itself. We hope that this project will help to reduce the processing time, number of anomalies in a transaction and of course the number of frauds that will be happening around the world.
3. Apart from this we are planning to add privacy, security and governance layers for the data. So, as we all know that the data in big data is of high importance and the security of it is also part of the project.

### **Team members and their roles :**

Prayag Nikul Purani :- Real-time data. Flink, Governance.

Sindhu Nagesha :- Kafka implementation, KSQLDB, Apache Superset.

Syed Faraaz Ahmed :- Streaming model implementation, Security, Elasticsearch.

Sai Vivek Chunduri :- Apache Spark, Privacy, Documentation (Shared work).

**Relevance to the course:**

This project is highly relevant to the course topics on stream processing and real-time data systems. It employs several key technologies and frameworks covered in the course, such as Kafka for real-time data streaming, Spark/Flink for stream processing, and Zookeeper for coordination. Additionally, the use of PostgreSQL for storing real-time data may align with the big data databases and warehouse concepts that will be discussed in the course.

**Technical Difficulty:**

Implementing a real-time Change Data Capture streaming project can present several technical challenges:

Data Consistency: Maintaining data consistency when capturing changes across multiple real-time systems can be difficult due to potential delays or failures.

System Performance: Real-time processing demands high-performance systems capable of handling large data volumes without performance degradation.

Data Security: Ensuring data security and privacy during real-time transfer between systems can be a challenge.

Integration with various technologies: Integrating diverse technologies like Docker, Postgres, Debezium, Kafka, and Apache Spark each with its own complexities, can lead to compatibility and version issues.

Scalability: Designing a scalable architecture that can handle increasing data loads without significant performance drops might be a key challenge.

**Novelty:**

The novelty of this real-time streaming project lies in its unique integration of diverse technologies like Docker, Postgres, Debezium, Kafka, Apache Spark to achieve real-time data streaming.A key novelty lies in its capacity to process and respond to data changes as they occur, a capability that sets it apart from traditional solutions. Furthermore, the project incorporates advanced error handling mechanisms to ensure data consistency and reliability, potentially leveraging novel techniques. Additionally, we may implement innovative strategies for scalability and performance optimization, further differentiating it from existing implementations in the real-time data streaming domain.

**Impact:**

The impact of this project can have a profound and wide-ranging impact across various domains:

Business Impact: By enabling real-time data processing, businesses can make more timely and data-driven decisions, leading to improved operational efficiency, enhanced customer service, and increased profitability.

Technical Impact: The project can contribute to the field of real-time data processing by showcasing the effectiveness of the integrated technology stack and novel techniques employed. It can serve as a starting point for similar initiatives in the future.

Societal Impact: By facilitating real-time processing of large volumes of data, the project can have broader societal implications. It can be applied in areas such as public safety (real-time surveillance and threat detection), healthcare (real-time patient monitoring), and environmental monitoring (real-time tracking of pollution levels).

**Heilmeier catechism:**

1. What are you trying to do? Articulate your objectives using absolutely no jargon.

The objective is to capture changes in data as they happen and share these changes with other systems in real-time.

1. How is it done today, and what are the limits of current practice?

Currently, many systems batch process data at regular intervals, which can lead to delays in data availability. The limitation is that these systems cannot react to changes in data in real-time.

1. What is new in your approach and why do you think it will be successful?

The new approach involves leveraging a combination of technologies to capture, process, and distribute data changes in real-time. This approach can be successful because it enables systems to respond and react to data changes as they occur, thereby enhancing the timeliness and relevance of the data.

1. Who cares? If you are successful, what difference will it make?

This real-time data streaming project holds significant relevance for businesses that depend on timely data for informed decision-making, such as financial institutions, e-commerce companies, and social media platforms. Successful implementation of this project can pave the way for improved decision-making processes, enhanced customer service experiences, and increased operational efficiency within these organizations.

1. What are the risks?

Potential risks associated with this project include the possibility of data loss during the transmission process, system performance degradation due to the high volumes of data being processed, and data security concerns arising from the real-time transfer of sensitive information.

1. How much will it cost?

The cost can vary depending on the chosen technology stack, volume of data being processed, required computational resources, and the specific cloud or software solutions employed, as these factors directly influence the expenses related to licensing, infrastructure, and scalability.

1. How long will it take?

The project's timeline is largely dependent upon factors such as its complexity, the available resources, and other variables, although the specific timeline has been outlined in the dedicated section above.

1. What are the mid-term and final “exams” to check for success?

Potential mid term check could involve the successful integration of the different technologies, demonstrating the real-time capture and processing of data, and meeting predetermined performance benchmarks. The final evaluation can be a successful real-world demonstration of the system, processing actual data in real-time.

### **Explain how the criteria in the rubric are met :**

| **Criteria** | **Explanations** |
| --- | --- |
| Abstract | N/A |
| Motivation | N/A |
| Literature survey | N/A |
| Methodology | Explained above. |
| Deliverables | Explained above |
| Team members and their roles | The workload is equally distributed between all the members of the group. |
| Relevance to the course | Many topics that will be covered in the class are used here. |
| Technical Difficulty | Integration of different software and version control. |
| Novelty | Input is not only from one type of data and apart from this we are implementing it with different technologies . |
| Impact | Improving the real-time data filtering and analysis. |
| Heilmeier catechism | It is briefly explained in the above paragraph. |