International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Implementing Low-Latency Data Streaming from SQL Server to BigQuery: A Kafka-Based Approach in Google Cloud Platform

Sainath Muvva

Abstract

This research paper presents an innovative approach to achieving near-real-time data synchronization between on-premise SQL Server databases and Google BigQuery, meeting the increasing need for timely analytics in contemporary data architectures. The proposed solution utilizes Apache Kafka as the central messaging system and Debezium as the Change Data Capture (CDC) connector, enabling low-latency data streaming with high scalability and resilience. Our architecture efficiently captures, processes, and transforms data before ingesting it into BigQuery, with updates reflected within an hour. The paper delves into the nuances of system design, implementation strategies, and optimization techniques, addressing challenges such as schema evolution, data consistency, and error handling. Through rigorous performance analysis and scalability testing, we demonstrate the efficacy of this Kafka-based approach in handling high-volume transactional data streams while leveraging BigQuery's powerful analytics capabilities. Additionally, the paper provides a detailed rationale for choosing Debezium as the CDC connector and Apache Kafka over alternatives like Google Cloud Pub/Sub, exploring the trade-offs and benefits of these technological choices. This comprehensive solution offers organizations a robust and cost-effective method to bridge on-premise databases with cloud platforms, facilitating advanced real-time analytics and data-driven decision-making in the cloud era.

Keywords: Debezium, Kafka, BigQuery, SQL Server, kafka topic, CDC

Introduction

In recent years, the demand for real-time data processing and analytics has grown exponentially across various industries, including finance, e-commerce, and digital media. This surge is driven by the need for immediate insights to support rapid decision-making and enhance customer experiences. As traditional batch processing methods fail to meet the requirements of modern data-driven applications, organizations are increasingly adopting streaming data architectures to enable continuous data flow between on-premise systems and cloud-based platforms.

This research introduces an innovative low-latency data streaming solution that integrates SQL Server with Google BigQuery using Apache Kafka and Debezium within the Google Cloud Platform (GCP) ecosystem. Our approach leverages Kafka as the central event streaming platform and employs Debezium as an efficient Change Data Capture (CDC) connector. This combination allows for near real-time capture and streaming of changes from SQL Server tables to Kafka topics, followed by processing, transformation, and ingestion into BigQuery for immediate analysis.

The proposed system addresses several critical challenges in modern data architectures, including minimizing data latency, ensuring fault tolerance, and seamlessly bridging on-premise relational databases

with cloud-based analytics platforms. By utilizing CDC with Debezium, our solution captures only incremental changes (inserts, updates, and deletes), significantly reducing the overhead associated with full data replication.

Our architecture is designed to handle high volumes of transactional data, offering scalability and reliability in large-scale environments. The focus on low-latency streaming enables businesses to maintain up-to-date data views in BigQuery with latencies as low as one hour, facilitating real-time reporting, dynamic dashboards, and accelerated decision-making processes.

This paper provides a comprehensive exploration of the Kafka-based streaming pipeline's design and implementation, highlighting the advantages of Debezium for CDC and the benefits of integrating SQL Server with BigQuery. We also delve into optimization strategies, error handling mechanisms, and performance considerations crucial for ensuring the solution's reliability and efficiency.

By offering a scalable and cost-effective approach to leveraging real-time analytics in the cloud, this research contributes valuable insights to the field of data engineering and provides organizations with a powerful tool for maintaining competitiveness in today's data-driven landscape.

Debezium

Debezium revolutionizes data synchronization as an open-source platform specializing in change data capture. It excels in real-time monitoring and streaming of database modifications across diverse systems. Compatible with a wide range of databases, Debezium seamlessly integrates with Apache Kafka, forming a robust foundation for event-driven architectures. By capturing changes at the source, it eliminates the need for periodic batch processes, enabling instant data updates across applications. This capability proves crucial in scenarios ranging from data warehousing to real-time analytics and microservices architectures. Debezium's strength lies in its ability to maintain data consistency in complex, distributed environments, making it a cornerstone of modern data infrastructure. It empowers organizations to build more responsive, agile systems that can adapt quickly to changing data. As businesses increasingly rely on real-time insights, Debezium's role in facilitating rapid data flow becomes ever more critical. Ultimately, it represents a significant leap forward in the evolution of data management, offering a scalable solution for the challenges of contemporary data-driven enterprises [1].

Functionality [1]:

1. Change Data Capture (CDC): Debezium reads changes from different databases (including Postgres) and exposes them as a stream of events for analytic systems.
2. Key Capabilities:

- Replay changes from the last n days if destination crashes
- Continue from where the master fell off during failover
- Selective table capture capability
- Parallel initial snapshot and streaming with resume capability
- Maintains transaction consistency
- Supports exactly-once semantics through state management

CDC

Change Data Capture (CDC) is a sophisticated software methodology that monitors and records modifications to data within a database system. By continuously processing database events in real-time

or near-real-time, CDC enables the efficient transmission of data updates. This approach is particularly valuable for scenarios requiring low-latency data replication and seamless cloud migrations without service interruptions. CDC excels at tracking various database operations, including insertions, updates, and deletions, providing a comprehensive view of data changes as they occur. This capability makes CDC an essential tool for organizations seeking to maintain data consistency across systems, facilitate timely analytics, and ensure robust data integration in dynamic environments [2].

Kafka

Apache Kafka is a distributed event streaming platform designed for high-throughput, fault-tolerant, real-time data handling. Originally developed by LinkedIn and now an open-source Apache project, Kafka excels in building real-time data pipelines and stream processing applications. It employs a publish-subscribe model, where producers send data to topics and consumers read from them. Kafka's distributed architecture allows for horizontal scaling, ensuring data availability even during server failures. Its ability to process millions of events per second makes it invaluable for big data applications requiring real-time processing and analysis [3].

Core Functionality:

- Distributed, scalable architecture
- High-throughput messaging
- Fault tolerance and data durability
- Stream processing capabilities
- Integration with various data systems

Key Use Cases:

- Real-time data streaming
- Event-driven architectures
- Data integration across platforms
- Log aggregation
- Microservices communication
- Real-time analytics
- Machine learning pipelines
- Data replication across data centers

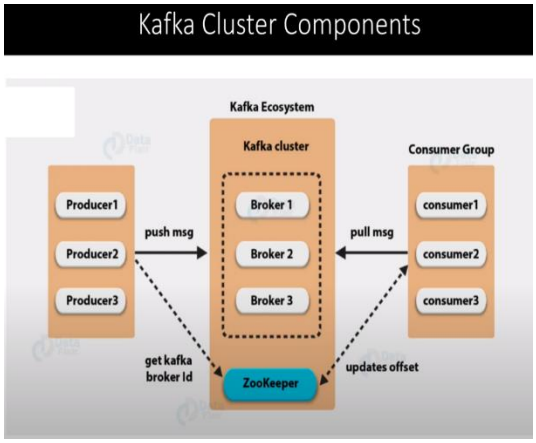

Fig 1. Architecture of Kafka

DataFlow

Dataflow is a fully managed Google Cloud service that enables unified stream and batch data processing at scale. It's built on Apache Beam, offering portability across different platforms and supporting multiple programming languages including Java, Python, and Go. The service automatically manages resources, provides autoscaling capabilities, and ensures exactly-once processing by default, while offering comprehensive monitoring through the Google Cloud console [4].

Use Cases:

1. Data movement - Ingesting or replicating data across subsystems
2. ETL workflows - Ingesting data into data warehouses like BigQuery
3. Powering BI dashboards
4. Applying ML in real-time to streaming data
5. Processing sensor data or log data at scale

Overall setup process

Setting up a near-real-time data pipeline requires integrating several crucial components to effectively capture and stream data. The process begins with establishing a Kafka Connect cluster, which acts as the backbone for synchronous data commitment to Kafka topics. Once this foundation is in place, Debezium is incorporated as a specialized connector within the Kafka Connect framework. This powerful tool is designed to detect and capture Change Data Capture (CDC) events from source databases, subsequently committing them to designated Kafka topics.

To enable Debezium's functionality with SQL Server, it's essential to activate CDC on the SQL Server instance using specific SQL commands listed below.

EXEC sys.sp_cdc_enable_db;

EXEC sys.sp_cdc_enable_table @source_object = '{sql_server_table}', @role = 'capture';

Following this activation, the Debezium SQL Server connector must be configured with the necessary connection details with respective to the sql server and target data specifications like topic name, polling interval and others.

Upon deployment of the Debezium connector to the Kafka Connect cluster, it initiates the streaming of change events into Kafka, where they are systematically committed to the specified topics. The Kafka Connect cluster then assumes the critical role of maintaining connector health and ensuring an uninterrupted flow of data from SQL Server to Kafka [5].

Establishing a comprehensive data pipeline, the next step after committing data to Kafka involves writing this data to Google BigQuery (GBQ). Dataflow, a managed service offered by Google Cloud Platform (GCP), provides a versatile solution for this task. Two primary methods can be employed to integrate Dataflow within this pipeline.

The first approach leverages the KafkaIO connector within a Dataflow template. This method involves packaging the Dataflow template using Docker and storing it in Google Cloud Storage. The template requires specific parameters, such as the Kafka IP address and topic name, and it can support JavaScript User-Defined Functions (UDFs) for custom data transformations.

The second approach utilizes Pub/Sub as an intermediary, employing Dataflow SQL. In this scenario, Kafka Connect is utilized to synchronize messages between Kafka and Cloud Pub/Sub. This includes installing the Cloud Pub/Sub connector, configuring the connector properties, and running Kafka Connect

as a service. Once the messages are flowing into Pub/Sub, a Dataflow SQL job is created to query the Pub/Sub topic and load the results into BigQuery.

Both methods necessitate the proper setup of service accounts and permissions. The pipeline operates as a streaming job, processing messages in real-time from Kafka to BigQuery, continuing indefinitely until manually stopped. The choice between the two methods depends on factors such as maintenance needs, scalability, transformation capabilities, and development complexity [6].

The implemented solution supports robust error handling through the use of dead-letter queues and accommodates various schema configurations, partitioning strategies, and transformation requirements. This results in a reliable, real-time data pipeline suitable for a wide range of use cases, including fraud detection, inventory management, and real-time recommendations.

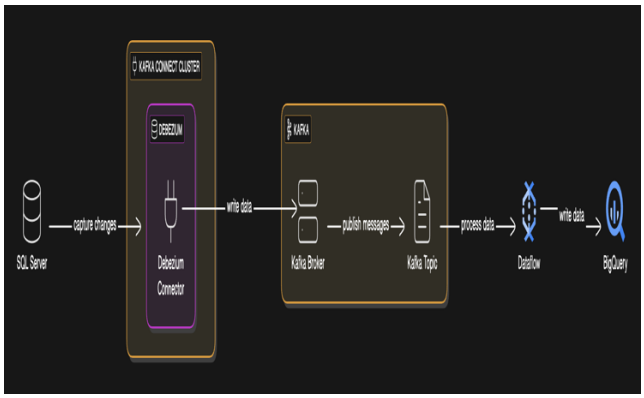

Fig 2. Architecture of Overall NRTDP

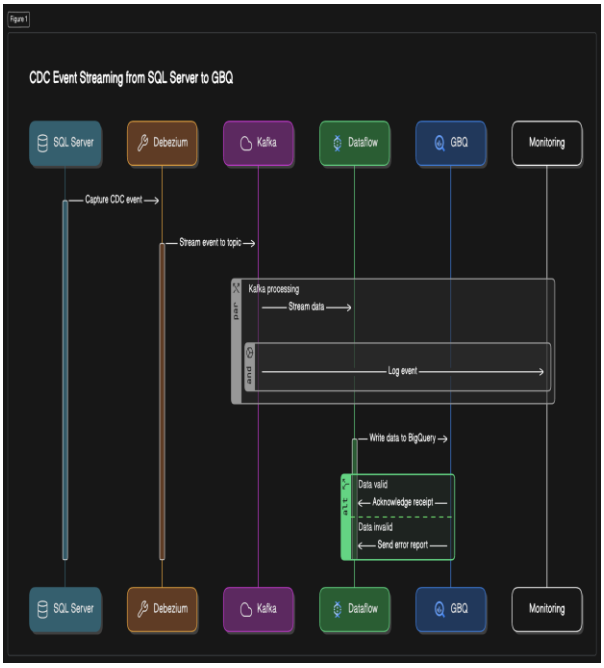

Fig 3. Sequence Diagram

Challenges:

When implementing a real-time data pipeline from Kafka to BigQuery via Dataflow, managing schema evolution presents significant challenges. These include handling dynamic schema changes in Kafka while

accommodating BigQuery's static schema requirements, ensuring data consistency and integrity across the pipeline, and maintaining backward and forward compatibility. Additional challenges involve error handling for schema mismatches, scaling schema management as the pipeline grows, mitigating latency impacts from continuous schema changes, and performing complex transformations to adapt to evolving data structures. To address these issues, solutions such as using schema registries, implementing schema versioning, leveraging Dataflow's transformation capabilities, and employing dead-letter queues for error handling are crucial for maintaining a robust and scalable pipeline that can handle the dynamic nature of real-time data streams.

Conclusion

The integration of SQL Server with BigQuery through Kafka, Debezium, and Dataflow creates a powerful framework for real-time data processing and analytics. While this comprehensive system offers adaptability and scalability, it requires addressing challenges to ensure smooth operation. These include managing evolving data structures, preserving data integrity, sustaining scalability, minimizing latency, and optimizing costs.

The pipeline begins with Debezium capturing Change Data Capture (CDC) events from SQL Server and publishing them to Kafka. These changes flow through Kafka Connect and enter Dataflow for real-time transformation and processing, before being loaded into BigQuery for analytics and reporting.

Key considerations involve adaptable schema management, robust error handling with message queues, and accommodating evolving data sources to maintain reliability and accuracy. Carefully managing BigQuery schema updates and data retention policies can help control costs while delivering current, real-time insights.

By addressing these challenges through appropriate configurations, monitoring, and optimization, the pipeline can deliver valuable real-time analytical capabilities. This empowers diverse business applications, from fraud detection to inventory management and personalized recommendations. The system's adaptability allows organizations to respond quickly to changing data, yielding high-quality insights and enabling data-driven decision-making.

Reference:

1. Sharath Gururaj, “Practical Notes in Change Data Capture with Debezium and Postgres”, https://medium.com/cermati-tech/practical-notes-in-change-data-capture-with-debezium-and-postgres-fe31bb11ab78
2. https://www.redhat.com/en/topics/integration/what-is-change-data-capture
3. R. Shree, T. Choudhury, S. C. Gupta and P. Kumar, "KAFKA: The modern platform for data management and analysis in big data domain," *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, Noida, India, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343593.
4. https://cloud.google.com/dataflow/docs/overview
5. Matheus Tramontini, “Simple CDC with Debezium + Kafka”, https://medium.com/nagoya-foundation/simple-cdc-with-debezium-kafka-a27b28d8c3b8
6. Thomas de Lazzari, “Kafka to BigQuery using Dataflow”, https://medium.com/google-cloud/kafka-to-bigquery-using-dataflow-6ec73ec249bb