Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

DataBay: A Unified Platform for Automating Data Warehouse Management, Real-Time Data Processing and Ensuring Quality

M. Ghadimi[*1], N. Baghayi[†2] and A. Shateri[‡1]

[1] Department of Engineering Sciences, University of Tehran, Tehran, Iran

[2] DataBurst.tech

ABSTRACT

As organizations increasingly depend on large-scale data for strategic decision-making, managing data warehouses has become a complex and resource-intensive challenge. This paper introduces DataBay, a unified platform designed to automate the entire data warehouse lifecycle, from data ingestion and transformation to real-time processing. The platform designed optimally to ensure faster implementation, more efficient data management high-performance data processing, real-time monitoring, storage efficiency, reliability, accuracy and scalablity. Through its seamless integration and flexibility, DataBay helps businesses make timely, data-driven decisions and enables continuous optimization of data workflows. This paper discusses the platform's architecture, its implementation in real-world industry settings and the significant business value it delivers.

Keywords: Data Warehouse, Real-time Data Processing, Data Management, Business Intelligence, Scalable Data Platform

ARTICLE INFO

Article history:
Research paper
Received 16, December 2024
Accepted 30, December 2024
Available online 30, December 2024

AMS subject classification: 68T09.

[*] Corresponding author: M. Ghadimi, Email: mostafa.ghadimi@ut.ac.ir
[†] niyusha.baghayi@databurst.tech
[‡] alireza.shateri@databurst.tech

1 Introduction

In today's data-rich environment, data-driven decision-making (DDDM) is vital for large companies' success. Transforming data through business intelligence (BI) and analytics turns it into a strategic asset that informs decisions at all levels. Comprehensive reports consolidate key data, enabling executives to make accurate and timely choices that shape a company's strategy and performance [1].

A data-driven culture allows businesses to swiftly respond to market changes, enhance efficiency, and improve customer experiences. Integrating analytics into decision-making leads to higher profitability, risk mitigation, and innovation [2]. Operational reports aid in supply chain management and cost optimization, while financial reports offer insights into economic health [3].

As companies increasingly rely on data to fuel their strategies, there is a growing demand for systems that can automate the collection, processing, and storage of this data. In particular, Change Data Capture (CDC) technologies, such as Kafka and Debezium, play a crucial role in ensuring that data is consistently and accurately moved from transactional systems to analytical environments in real-time.

This article explores the importance of data-driven decision-making, discusses the role of CDC in automating data workflows, and highlights the need for modern data architectures that can handle the demands of big data. We will also discuss how distributed systems and microservices are critical in overcoming the limitations of traditional data architectures, addressing the challenges of scaling and reliability.

2 The Evolution of Data Management Architectures

With the advent of the internet, websites transitioned from being static and newspaper-like to becoming dynamic, allowing user interactions to shape the data they produced. This shift marked a significant change in the architecture of web applications, as they not only collected data explicitly entered by users (such as usernames and passwords) but also gathered data from user interactions. This dynamic interaction data became crucial for organizations to make informed, data-driven decisions [4].

As internet usage expanded, the concept of "big data" emerged. This term refers to large volumes of user-generated data that exhibit certain characteristics, collectively known as the 6Vs of Big Data: Volume, Velocity, Variety, Veracity, Value, and Variability. These characteristics highlight the challenges organizations face in managing, processing, and extracting value from massive, fast-moving, and diverse datasets [5].

The increase in data volume posed significant challenges for traditional data architectures, especially when it came to scaling up. Single-node systems, with their inherent hardware limitations, were not designed to handle the exponential growth in data. This is where Moore's Law, which suggests that computing power doubles approximately every two years, comes into play. While Moore's Law has spurred growth in processing capabilities, the sheer scale of data demands architectures that go beyond single-node systems. The need for distributed systems and

microservices to manage this data became clear, enabling more efficient and flexible handling of vast data flows [6].

Big Data and the Need for Modern Data Storage Technologies

As businesses began to grapple with big data, they needed new technologies to store and process this information. Traditional OLTP (Online Transaction Processing) databases were primarily used to handle transactional data but were not optimized for complex analytical queries. In dynamic environments, these databases served their purpose well for transactions but fell short when it came to providing insights from vast data sets that demanded extensive analysis [7].

In 1990, William Inmon introduced the concept of the data warehouse, designed to provide a centralized repository that stored historical data from multiple transactional systems. The data warehouse became a foundational element of business intelligence (BI) systems, offering a single source of truth that organizations could rely on for their analytical needs [8].

As the volume and complexity of data continued to grow, other models emerged to address the need for more scalable and flexible data architectures. These include Data Lakes, Data Lakehouses, and Data Mesh, each offering a unique approach to managing data depending on its structure and use case. This article focuses primarily on the Relational Data Warehouse and Modern Data Warehouse models, which serve as critical components of today's data infrastructure [9][15].

Moore's Law and the Shift to Distributed Data Systems

While Moore's Law predicts exponential growth in processing power, it also reveals the limits of single-node systems in handling big data. As the need for scalability and reliability increased, businesses started adopting distributed systems and microservices architectures. These approaches enable companies to distribute the processing and storage of data across multiple nodes, allowing for more efficient handling of large-scale data sets while maintaining high availability and minimizing the risk of failure from a single point [10][19].

Distributed architectures have become essential in the world of big data. They provide the flexibility needed to scale as data volumes grow, and they offer fault tolerance by ensuring that the system remains operational even if one or more components fail. This has allowed businesses to handle real-time streaming data at scale, which is crucial for supporting data-driven decision-making in fast-paced business environments [11].

The Role of CDC and Data Integration

As data became more complex and voluminous, the need for real-time data integration grew. Change Data Capture (CDC) technologies, such as Kafka and Debezium, provide a scalable solution for streaming data changes from transactional systems to data warehouses in real time. By capturing changes directly from the transaction logs of databases, CDC ensures that updates, deletes, and inserts are immediately reflected in downstream systems, maintaining the consistency and accuracy of data.

The integration of CDC technologies with distributed systems allows for seamless data flow across multiple platforms, enabling businesses to make timely and informed decisions based on the most current data. These technologies also ensure that the data is synchronized across systems, eliminating discrepancies that could arise from outdated or inconsistent information. This level of integration has proven essential for businesses operating in data-intensive environments, such as e-commerce, where logistical decisions need to be made based on real-time data [12][13].

3 Data Quality and Its Pillars in Real-Time Systems

The quality of data is critical in ensuring that business decisions are based on accurate and actionable information. Data quality can be assessed using six key pillars: accuracy, completeness, consistency, timeliness, validity, and uniqueness. These pillars are essential for maintaining the integrity of data, especially in environments where real-time processing is a necessity [14].

Debezium plays a crucial role in maintaining data quality in real-time systems. By capturing database changes directly from the transaction logs, it ensures that all updates, deletions, and inserts are immediately reflected in downstream systems. This minimizes the risk of errors or inconsistencies and ensures that businesses can rely on up-to-date and accurate data for decision-making [17].

In addition, CDC technologies like Debezium ensure that data is synchronized across systems in real time, helping to maintain data consistency and reducing the risk of discrepancies. Real-time synchronization also helps to preserve the validity of data by ensuring that all changes comply with defined business rules and data constraints [13].

Data Warehouse Architectures: From OLTP to OLAP

To process large volumes of data efficiently, businesses transitioned to OLAP (Online Analytical Processing) databases, which are specifically designed for read-heavy analytical workloads. Unlike OLTP databases, which are optimized for transactional processes, OLAP databases use columnar storage and pre-aggregated data to support complex queries on large datasets [18].

The architecture of OLAP databases is more suited for analytical purposes because it allows for faster querying of historical data. This is particularly useful for data marts, which are subsets of data warehouses that focus on specific business areas such as sales, finance, or marketing. The combination of OLAP databases and data marts enables businesses to perform deep analytical queries without compromising the performance of operational systems [18].

Key Requirements for a Data Warehouse/BI System

Incorporating real-time data integration through tools like CDC with Debezium offers numerous advantages, including reduced latency, improved data consistency, and scalability. However, to ensure the success of a DW/BI system that integrates real-time data, businesses must adhere to several fundamental requirements. Kimball and Ross (2022) outline these key requirements, which remain critical for a successful data architecture [12].

Simplicity and Speed: A DW/BI system must make information easily accessible. The contents of the system should be understandable to business users, not just developers. The data structures and labels must mirror the users' thought processes and terminology. Business users need to manipulate data in endless combinations to derive insights. Therefore, the business intelligence tools that access this data must be simple to use and return query results quickly. This requirement can be summarized as "simple and fast." If users can access actionable insights in real-time with minimal complexity, they will be more likely to adopt and trust the system [12].

Consistency: A DW/BI system must present information consistently. The data it houses must be credible, meaning that it must be assembled from various sources, cleaned, and quality-assured before being made available for analysis. Furthermore, consistency also implies that common labels and definitions are used across different data sources. If two performance metrics share the same name, they must represent the same thing; if they differ, they must be distinctly labeled. This approach ensures that the data remains consistent across the system and that users can trust the results of their analyses [12].

Adaptability: A DW/BI system must be designed to adapt to change. This includes changes in user needs, business conditions, and technology. Since user requirements and business environments evolve, the system must be flexible enough to incorporate new data and adjust to changes without invalidating existing data or disrupting current operations. If modifications to the descriptive data in the system are required, these should be transparently managed, ensuring the changes do not disrupt the workflow or data accuracy [12].

Timeliness: As the DW/BI system is used more intensively for operational decisions, raw data must be converted into actionable insights quickly. Businesses need to have realistic expectations regarding how fast they can process data. This is particularly important for real-time applications where data needs to be analyzed within hours, minutes, or even seconds, enabling rapid decision-making. Therefore, real-time data integration, such as through CDC, is crucial for providing timely, reliable insights [12].

Security: The DW/BI system must act as a secure bastion to protect organizational data. Since these systems store critical business information, such as sales data, pricing information, and customer data, it is crucial to control access effectively. Unauthorized access to this data can lead to severe consequences for businesses, making data security a top priority [12].

Trustworthy Foundation for Decision-Making: The DW/BI system must serve as the authoritative source for decision-making. This system must house reliable data that supports the decisions of key stakeholders. Ultimately, the value of a DW/BI system is measured by its ability to inform decisions that drive business success. The data warehouse becomes the backbone of a decision support system when it provides accurate and up-to-date information for strategic choices [12].

User Acceptance: Finally, for a DW/BI system to be deemed successful, it must be accepted and used by the business community. Unlike operational systems, where employees have no choice but to use the system, the adoption of a DW/BI system is often optional. To gain acceptance, the system must offer business users a simple and fast way to access actionable insights. If the system meets the needs of users and delivers tangible value, it will gain traction across the organization [12].

4 Methodology

Creating, designing, and maintaining data warehouses in organizations is a costly process both in terms of time and technical complexity, especially considering the variety of available tools. As data engineering teams operate between critical teams such as software engineering (upstream) and data analytics, data science, and business intelligence teams (downstream), effective communication and collaboration with other teams is essential to implement business requirements. Given that communication is inherently time-consuming, one of the primary requirements we addressed in this paper was to provide data engineers with a platform that abstracts the technical complexity of the system as much as possible. We developed a pluggable platform that not only does not impose technical limitations on data engineers but also enables them to meet their needs—creating, designing, and maintaining data warehouses—more efficiently with fewer specialists involved.

Data Warehouse Construction as a Single Source of Truth

To build a data warehouse as a single source of truth, it is necessary to gather all data from various data sources and prepare it for use. The data model must be designed in a way that allows downstream users to easily access the data with high performance. Furthermore, depending on the business requirements, data must be stored with specified granularity in a historical format.

(Note: One of the primary differences between a data warehouse and an operational data store is that in an operational data store, data is used in a transactional manner, while in a data warehouse, data is stored in a historical format.)

The first step in creating a data warehouse is reading data from different data sources. From a time perspective, data can be obtained in two ways: in real-time or in batches. A major issue with batch processing is that if any changes occur in the data in an OLTP database, these changes may be lost due to the delay between reading different chunks of data, thus making historical data inaccessible.

On the other hand, one of the key challenges in production databases is ensuring that the process of reading data does not negatively impact database performance. One effective solution to detect changes in databases is the concept of Change Data Capture (CDC). Various tools have been developed for CDC, with Debezium being one of the most well-known. Debezium is an open-source tool that helps us capture data from data sources and stream it to Kafka without affecting the performance of production databases. Debezium supports reading data from most OLTP databases such as PostgreSQL, MySQL, SQL Server, and others.

In our platform, we use Debezium to capture data from OLTP databases, stream it into Kafka, and store it temporarily based on the retention period defined for the Kafka topic. Debezium detects row-level changes, where each CRUD operation results in sending corresponding records to Kafka. Additionally, using Single Message Transformation (SMT), we apply transformations such as flattening on each record before sending it to Kafka.

Data Serialization: Avro vs. JSON

In our platform, we utilize Avro as the primary serialization format for streaming data through Kafka. While Debezium supports both JSON and Avro, we chose Avro primarily due to its superior performance with large-scale datasets. JSON is human-readable and flexible but is less efficient when dealing with real-time, high-volume data processing environments.

We conducted performance benchmarks comparing Avro and JSON in terms of data size, throughput, and serialization/deserialization speeds. These benchmarks were performed using two dedicated servers with the following configuration:

- CPU: Two Intel E5-2660 v2 10-core CPUs at 2.20 GHz
- RAM: 256GB ECC Memory (16x 16 GB DDR4 1600MT/s dual rank RDIMMs)
- NIC: Dual-port Intel 10Gbe NIC (PCIe v3.0, 8 lanes)

The TPCH Benchmark, specifically the "orders" table, was used for all experiments. The results of these benchmarks strongly favored Avro due to its binary format and schema-based serialization [20].

5 Experiment 1: Data Transfer Comparison

In the first experiment, we transferred specific data volumes across a network between two separate nodes. Avro achieved significantly smaller data sizes, as it uses a binary format compared to the text-based JSON format. Avro reduced the data transferred by up to one-third compared to JSON. This reduction in data size directly resulted in higher throughput, with Avro achieving up to three times greater throughput than JSON. However, as the number of rows increased, the throughput started to plateau due to network link capacity, with the throughput ratio between Avro and JSON stabilizing around 2.5:1.

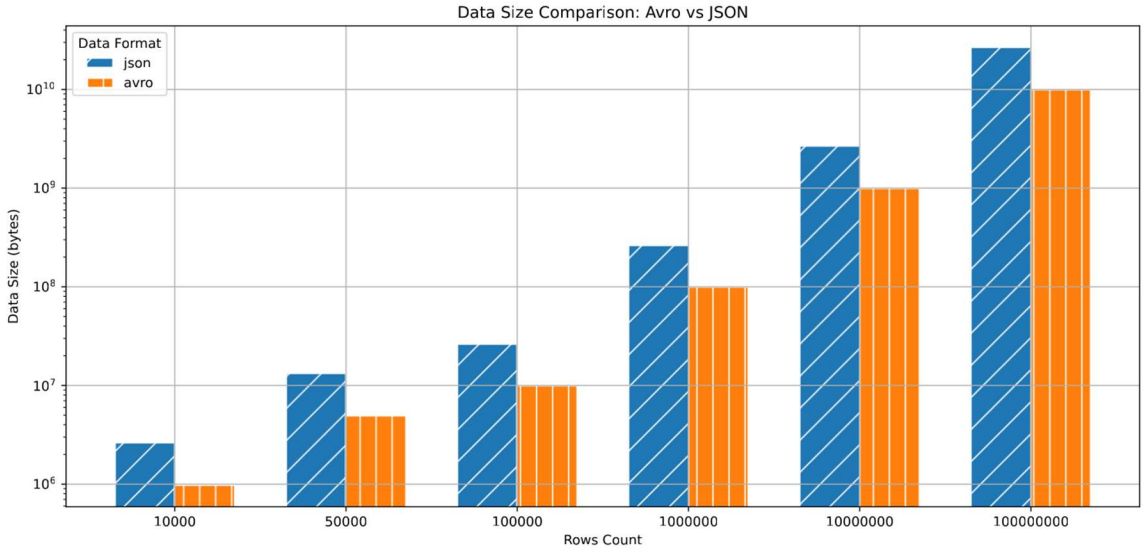

Figure 1 Data Size Comparison: Avro reduced the data transferred by up to one-third compared to JSON [21]

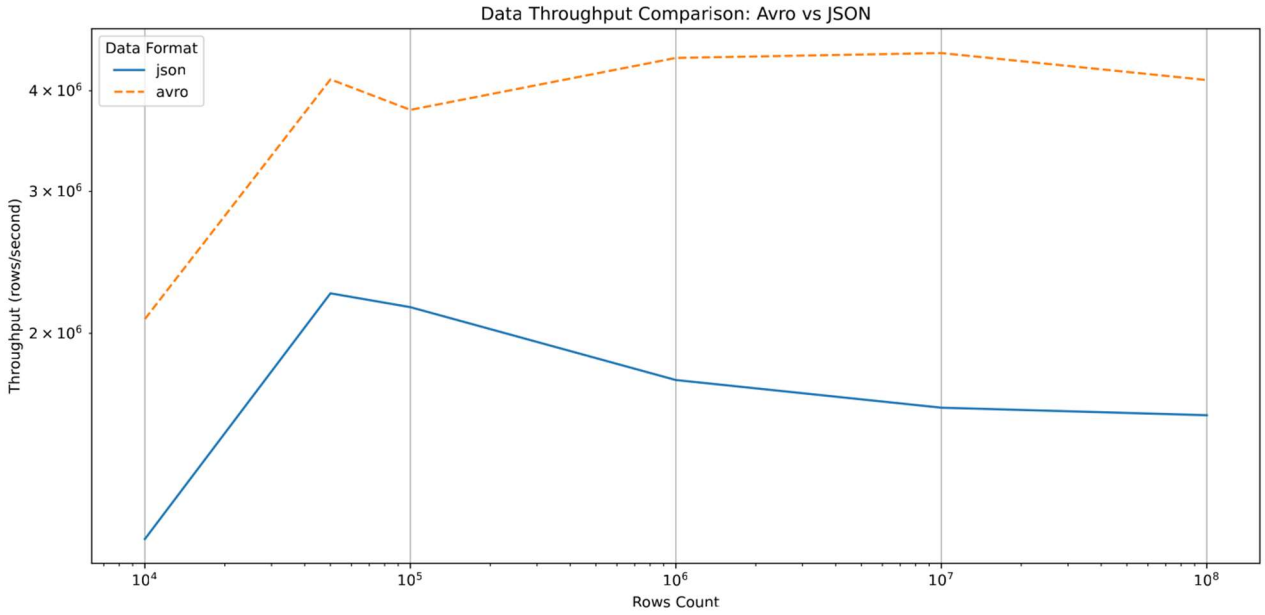

Figure 2 Data Throughput Comparison: Avro achieved throughput up to three times higher than JSON, but the ratio approached 2.5 as the row count increased [21]

6 Experiment 2: Serialization/Deserialization Performance Comparison

In the second experiment, we compared the serialization and deserialization performance of Avro and JSON using the fastest C++ libraries for both. We used RapidJson and Avro's official library for C++ in implementation. The results indicated that Avro outperforms JSON significantly in both serialization and deserialization. Avro was able to serialize up to 400,000 rows per second, while

JSON could serialize only 80,000 rows per second. In deserialization, Avro achieved an impressive 450,000 rows per second, whereas JSON reached only 26,000 rows per second.

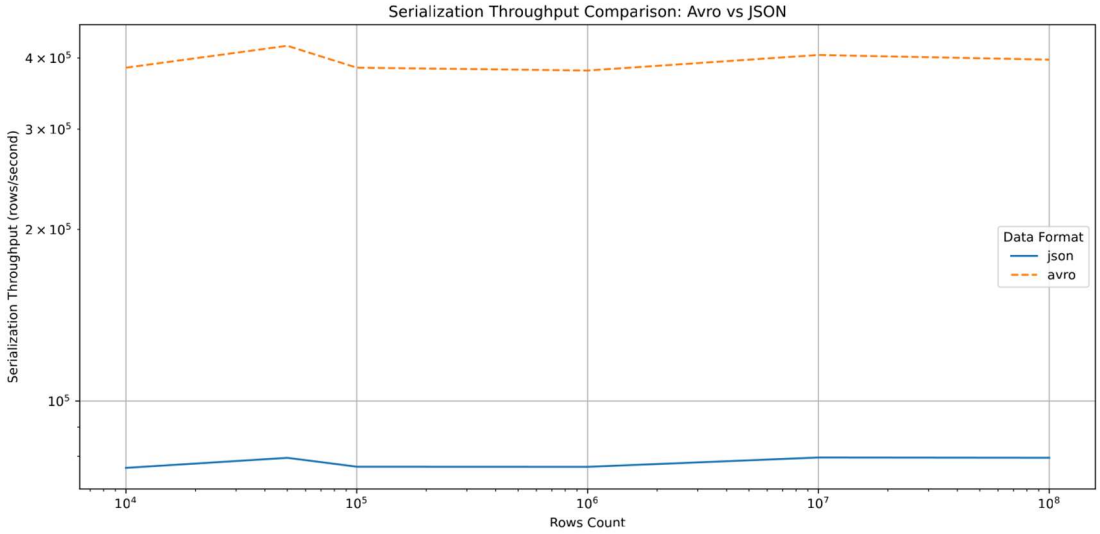

Figure 3 Serialization Performance Comparison: Avro processes 400,000 rows per second, significantly faster than JSON, which processes only 80,000 rows per second [21].

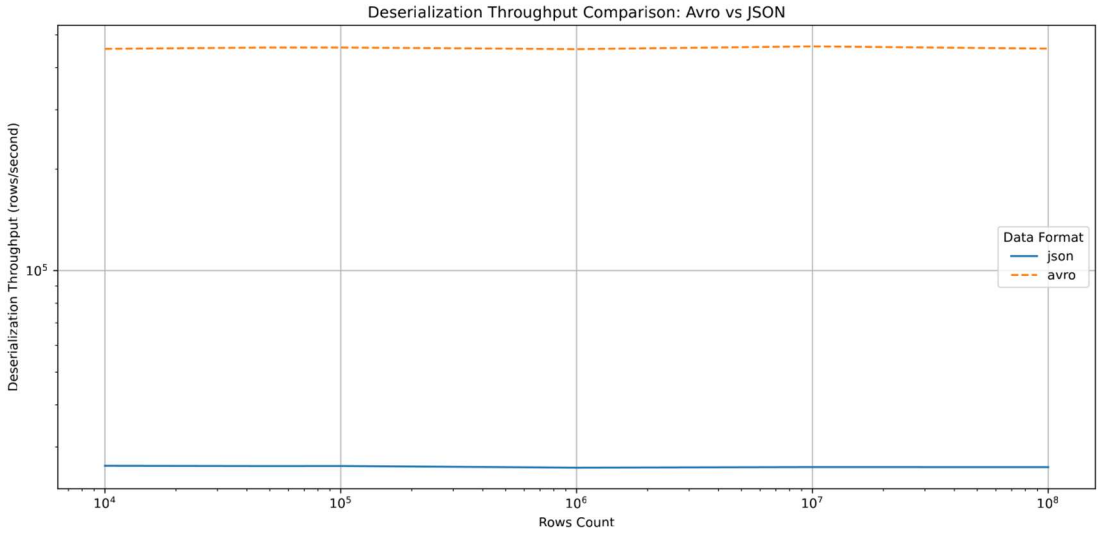

Figure 4 Data Throughput Comparison: Avro achieved throughput up to three times higher than JSON, but the ratio approached 2.5 as the row count increased. Deserialization Performance Comparison: Avro deserializes 450,000 rows per second, while JSON handle

This benchmark data strongly influenced our decision to use Avro as the default format for data serialization in our platform. It provided significant advantages in terms of data transfer efficiency, throughput, and serialization/deserialization performance, all of which are critical for real-time, high-volume data integration.

Code Automation for Efficient Connector and Transformation Management

The process of building connectors, converting data types to formats compatible with the sink database, and creating tables can be highly time-consuming. In this platform, we have developed a code that automatically generates connectors, reads data types from the source database, adjusts

them according to the sink database's data type support, and automatically manages the implementation. This tool prevents errors like conflicts that may arise from recreating connectors. If any issues occur during parsing messages by a connector worker, the problematic messages are sent to a Dead Letter Queue (DLQ) for review.

One of the significant advantages of using Debezium and Kafka is that they leverage the Log Sequence Number (LSN) to track the data changes. In case of a failure in a connector, the system can resume data processing from the point of failure when the connector is reactivated.

Kafka's Publisher-Subscriber architecture (also known as Producer-Consumer) helps decouple system components, making maintenance easier. Additionally, Kafka uses the Raft consensus algorithm as per KIP-595, eliminating the need for additional tools like Zookeeper for cluster and metadata management. This architecture also makes Kafka highly scalable and fault-tolerant, allowing for the creation of highly available broker clusters.

Data Transformation and Masking

After the data is stored in the staging layer, transformation rules are defined in YAML files. These transformations, based on business discussions with product managers and business leaders, can vary by industry and domain. To simplify the process and reduce technical complexity for users, we write transformations in human-readable YAML files that can be easily modified.

Transformation logic can be executed using Python libraries such as PySpark, Pandas, or Polars depending on the volume of data. In addition to data cleaning, some sensitive data might need to be masked. This is handled by configuring the necessary settings in the transformation YAML file.

Once the data is in the staging layer, heavy transformations are processed using Spark pipelines, and the final data is stored in the core layer (data warehouse). The staging layer ideally uses OLTP databases, while OLAP databases can be used in the core layer of the data warehouse, depending on team needs.

Data Storage: OLAP and OLTP

Commonly used data architectures store data in star schema format in OLTP databases and then process it into wide or flat tables in OLAP databases. These flat tables do not require joins, making them efficient for querying. Columnar databases like ClickHouse are well-suited for this type of architecture, as they store columns in Sorted String Tables (SSTables), which are highly efficient for scenarios where only a subset of columns is needed in queries.

Monitoring and Quality Assurance

Once data is processed, it is essential to monitor the health of the system. We use Prometheus to collect and store metrics for every tool in the architecture. For tools like Kafka and Debezium, which use Java codebases, we convert JVM metrics into a format compatible with Prometheus.

For other tools, exporters are used to expose metrics. These metrics are scraped by Prometheus at defined intervals and stored in its time-series database.

We use Grafana for visualizing the metrics and setting up alerts. If any tool's performance degrades, alerts are sent through Alert Manager to systems like Slack or Discord for notification.

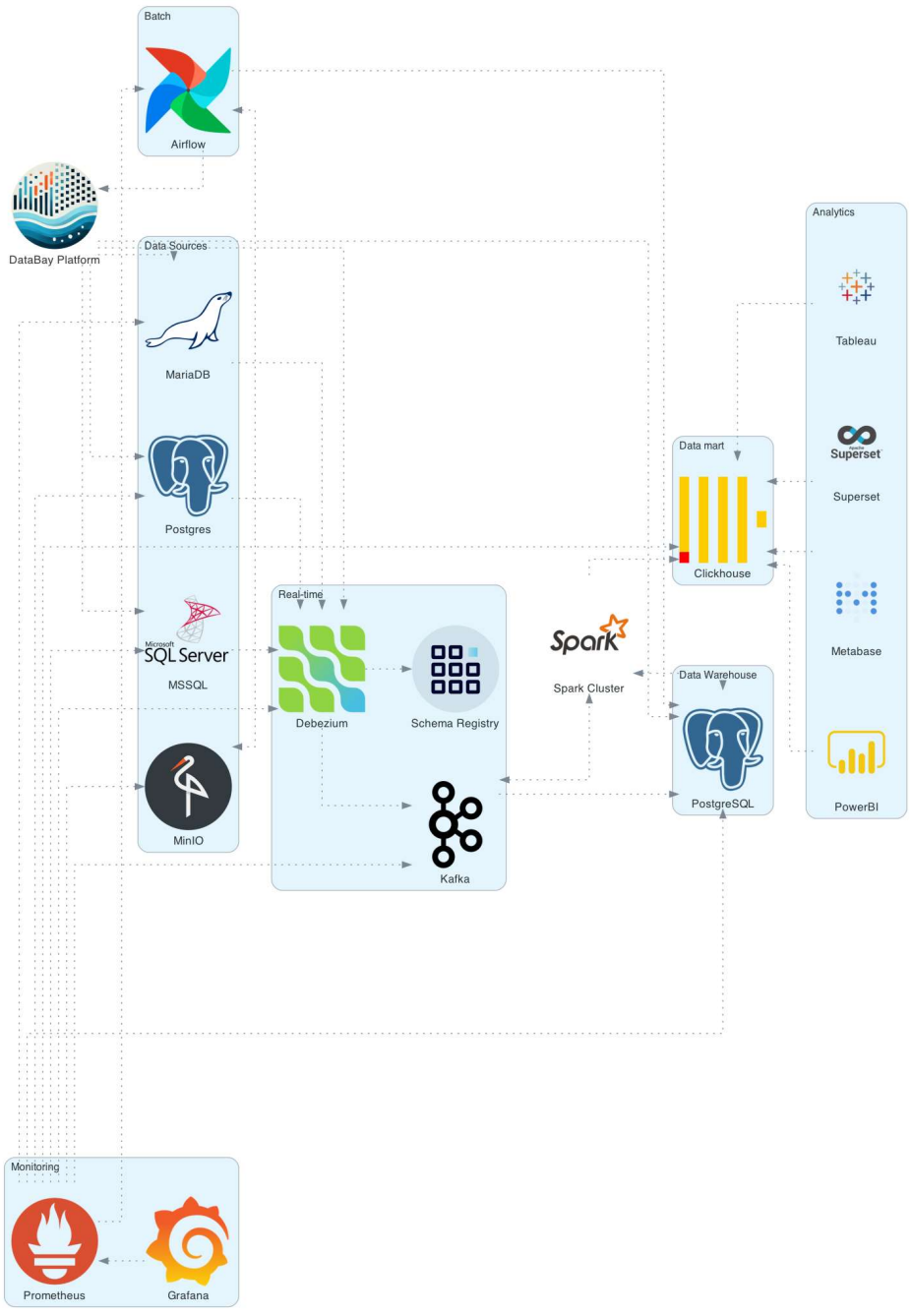

Figure 5 Shows the monitoring stack and key components of our platform, including Debezium, Kafka, and Prometheus, along with the flow of data and monitoring metrics. The diagram provides a high-level overview of how each component integrates into the platform to ensure performance and data quality [21]

For more detailed information and access and access to the source code and diagrams, please refer to the GitHub repository.

Data Quality Evaluation

Data quality is essential throughout the transformation process, and it is assessed in terms of the following six key dimensions:

- Accuracy: Data must accurately represent the real-world entities or phenomena it models.
- Completeness: Ensure that all necessary data is present, handling missing or null values appropriately.
- Consistency: Data must be consistent across systems and datasets.
- Timeliness: Data should be up-to-date and available when needed.
- Uniqueness: Data should not contain duplicates.
- Validity: Data must meet defined formats and acceptable ranges.

These checks are automated using Airflow for pipeline orchestration and Spark for processing. If any issues arise, alerts are triggered, and the problem is isolated using Kubernetes Pod Operators in Airflow to ensure that the pipelines are decoupled and operate independently.

Finally, after modeling the data using YAML files, setting up the pipelines, and monitoring the performance of tools and data quality, the data needs to be stored in a Data Mart to make it suitable for downstream use. To achieve this, once the source and destination databases are selected, the required tables are automatically defined, along with the appropriate primary keys for sorting SSTable blocks in the background and ensuring faster data access. These are all the steps involved in the design and implementation of the data platform we have developed.

7 Conclusion

In this paper, we presented a unified platform designed to streamline the creation, management, and maintenance of data warehouses in modern enterprises. By automating critical processes such as data ingestion, transformation, and monitoring, the platform significantly reduces the complexity and resource demands typically associated with data warehouse management. Through the integration of advanced tools like Debezium for Change Data Capture and Kafka for real-time data processing, we ensure high performance and minimal disruption to production environments.

The decision to use Avro over JSON for data serialization, backed by benchmark comparisons, highlights the practical considerations of performance and scalability in real-world applications. This choice, coupled with automation in connector management and data transformation, allows data engineers to achieve faster implementation and more efficient workflows, even with minimal specialized expertise.

The platform has proven its effectiveness in large-scale, real-world environments, demonstrating the potential for modern, automated data infrastructure to drive faster, data-driven decision-making. By abstracting technical complexity and optimizing data workflows, the platform offers businesses a scalable, flexible solution to manage growing data demands and stay competitive in an increasingly data-centric world.

References

[1] A. N. Turi, "data-driven-decision-making-in-digital-entrepreneurship," International Journal of Industrial and Systems Engineering, vol. Vol:16, No:4, 2022, Jul. 2022, https://www.researchgate.net/publication/361778286_data-driven-decision-making-in-digital-entrepreneurship

[2] T. H. Davenport, "Competing on Analytics," ResearchGate, 2006, Available: https://www.researchgate.net/publication/7327312_Competing_on_Analytics

[3] R. Sharda, Dursun Delen, and Efraim Turban, "Business Intelligence, Analytics, and Data Science: A Managerial Perspective," Jan. 23, 2017. https://www.researchgate.net/publication/318456888_Business_Intelligence_Analytics_and_Data_Science_A_Managerial_Perspective

[4] D. Dwyer, "The Recent History of the World Wide Web | Insights," Inspire.scot, Aug. 31, 2015. https://www.inspire.scot/blog/2015/08/31/the-recent-history-of-the-world-wide-web190 (accessed Dec. 05, 2024).

[5] G. Jha, "Understanding the 6 Vs of Big Data: Unraveling the Core Characteristics of Data-Driven Success," Medium, Oct. 2024. https://medium.com/@post.gourang/understanding-the-6-vs-of-big-data-unraveling-the-core-characteristics-of-data-driven-success-38e883e4c36b

[6] J. Reis and M. Housley, Fundamentals of Data Engineering. "O'Reilly Media, Inc.," 2022.

[7] J. Schaffner, A. Bog, J. Krüger, and A. Zeier, "A Hybrid Row-Column OLTP Database Architecture for Operational Reporting," Business Intelligence for the Real-Time Enterprise, pp. 61–74, 2009, doi: https://doi.org/10.1007/978-3-642-03422-0_5.

[8] M. V. Mannino and Z. Walter, "A framework for data warehouse refresh policies," Decision Support Systems, vol. 42, no. 1, pp. 121–143, Oct. 2006, doi: https://doi.org/10.1016/j.dss.2004.11.002.

[9] B. wong, "Navigating the Data Architecture Landscape: A Comparative Analysis of Data Warehouse, Data Lake, Data Lakehouse, and Data Mesh," Oct. 2023, doi: https://doi.org/10.20944/preprints202309.2113.v1.

[10] Ion, "For Big Data, Moore's Law Means Better Decisions," AMPLab - UC Berkeley, Feb. 08, 2013. https://amplab.cs.berkeley.edu/for-big-data-moores-law-means-better-decisions.

[11] S. Mazumder, R. Singh Bhadoria, and G. C. Deka, Eds., Distributed Computing in Big Data Analytics. Cham: Springer International Publishing, 2017. doi: https://doi.org/10.1007/978-3-319-59834-5.

[12] F. M. Imani, L. Widyasari, and Satria Perdana Arifin, "Optimizing Extract, Transform, and Load Process Using Change Data Capture," 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Dec. 2023, doi: https://doi.org/10.1109/isriti60336.2023.10468009.

[13] Alperen Sayar, Ş. Arslan, Tuna Çakar, Seyit Ertuğrul, and Ahmet Akçay, "High-Performance Real-Time Data Processing: Managing Data Using Debezium, Postgres, Kafka, and Redis," 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–4, Oct. 2023, doi: https://doi.org/10.1109/asyu58738.2023.10296737.

[14] Sandeep Rangineni, Amit Bhanushali, M. Suryadevara, and Kiran Peddireddy, "A Review on Enhancing Data Quality for Optimal Data Analytics Performance," International Journal of Computer Sciences and Engineering, vol. 11, no. 10, pp. 51–58, Oct. 2023, doi: https://doi.org/10.26438/ijcse/v11i10.5158.

[15] J. Serra, Deciphering Data Architectures. 2024.

[16] M. Kleppmann, Designing data-intensive applications : the big ideas behind reliable, scalable, and maintainable systems. Sebastopol, Ca: O'reilly Media, 2018.

[17] J. L. Wiltz, B. Lee, R. Kaufmann, T. J. Carney, K. Davis, and P. A. Briss, "Modernizing CDC's Practices and Culture for Better Data Sharing, Impact, and Transparency," Preventing chronic disease, vol. 21, Mar. 2024, doi: https://doi.org/10.5888/pcd21.230200.

[18] S. S. Conn, "OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis," Proceedings. IEEE SoutheastCon, 2005., doi: https://doi.org/10.1109/secon.2005.1423297.

[19] "Grokking Concurrency," Manning Publications, 2023. https://www.manning.com/books/grokking-concurrency (accessed Aug. 26, 2024).

[20] "TPC BENCHMARK TM H." Accessed: Dec. 05, 2024. [Online]. Available: https://www.tpc.org/TPC_Documents_Current_Versions/pdf/TPC-H_v3.0.1.pdf

[21] data-burst, “GitHub - data-burst/databay-paper-assets,” GitHub, 2024. https://github.com/data-burst/databay-paper-assets (accessed Dec. 05, 2024).