Open Source Data Architecture For Real-time Processing

LaxmiPriya Kotha
CS652 Big Data Architecture
Sacred Heart University
Fairfield, CT USA
kothal2@mail.sacredheart.edu

***Abstract*—In this paper, we present a comprehensive data architecture designed to handle real-time data processing, storage, and analytics using various open-source platforms. The architecture integrates multiple components including MySQL, Apache Cassandra, Apache Kafka, Metabase, Apache Spark, Apache Atlas, Docker, Apache Hive, Apache Druid, Pandas (Python), and Superset to ensure data consistency, scalability, and effective data governance. The proposed architecture leverages Apache Atlas for master data management, ensuring robust metadata management and data governance.**

Keywords— Data Architecture, Real-Time Data Processing, Data Governance, Metadata Management, Data Storage Data Analytics, Open-Source Tools, Data Integration, Scalability, Apache, Atlas, Apache Kafka Apache Cassandra, Apache Hive, Apache Druid, Docker, Metabase, Apache Spark, Pandas (Python), Superset, Data Wrangling

I. INTRODUCTION

In today's digital era, the exponential growth of data demands robust architectures that can efficiently and effectively manage vast datasets. Our proposed architecture meets this challenge by integrating a suite of open-source tools, each tailored to specific aspects of data handling, including storage, processing, analytics, and visualization. This paper details the components of this architecture, their functions, and the overall advantages of the system, with an emphasis on the integration of Apache Atlas for master data management. Apache Atlas plays a crucial role in ensuring consistent data governance and comprehensive metadata management across the entire architecture.

II. BACKGROUND INFORMATION

A. Data Providers

MySQL is chosen as the primary data provider for its reliability, ease of use, and strong community support. It excels in web-based applications due to its speed and ease of integration with various web technologies. MySQL ensures data consistency, integration with other components, scalability, real-time data processing, and cost-effectiveness.

B. Master Data Management

Apache Atlas**:** Apache Atlas is integrated into the architecture to provide robust data governance and metadata management capabilities. It ensures consistent and accurate data across the entire architecture by offering detailed metadata management and lineage tracking. Apache Atlas's integration with other components in the Hadoop ecosystem makes it a valuable addition for maintaining data quality and compliance.

C. Data layer Components

1.APIs (REST API):

REST APIs (Representational State Transfer Application Programming Interfaces) are integral to our architecture for enabling seamless data access and manipulation. REST APIs are renowned for their simplicity and stateless operations, which means each request from a client to a server must contain all the information needed to understand and process the request. This statelessness ensures that each interaction remains independent, reducing server-side complexity and improving scalability. Additionally, REST APIs support a wide range of languages and frameworks, which makes them highly versatile and easily integrable into diverse systems. Their adherence to standard HTTP methods (GET, POST, PUT, DELETE) ensures that they can handle real-time capabilities effectively, facilitating prompt and secure data exchanges. This approach not only enhances usability but also fortifies the architecture's security, as REST APIs can be fortified with various authentication mechanisms such as OAuth, JWT, and API keys to safeguard data transactions.

2.Read-Only Data Stores (Apache Cassandra):

Apache Cassandra is chosen as the read-only data store for its exceptional scalability and high availability. Its architecture is designed to handle large volumes of data across multiple nodes without any single point of failure, making it ideal for distributed environments. Cassandra's write-optimized design ensures that it can handle high write throughput with low latency, which is critical for applications requiring durable and persistent storage solutions. The data stored in Cassandra is immutable, meaning it cannot be altered once written, ensuring data integrity and reliability. Its ability to store data passively and persistently at high volumes allows it to serve as a robust backbone for applications requiring constant and reliable data access. Furthermore, Cassandra supports multi-data centre replication, which enhances data availability and disaster recovery capabilities, ensuring that data remains accessible even in the event of node or network failures.

3.Streaming Channels (Apache Kafka):

Apache Kafka serves as the streaming channel in our architecture, providing a platform for real-time data streaming and messaging. Kafka's design principles emphasize durability, scalability, and low latency, which are crucial for high-throughput data streaming applications. Kafka operates as a distributed publish-subscribe messaging system, where data is published to topics and can be consumed by multiple subscribers in real time. This architecture supports the seamless ingestion of large volumes of event data, ensuring that messages are processed in a reliable and fault-tolerant manner. Kafka's durability is achieved through its persistent log, where all messages are written to disk, ensuring that data is not lost even if consumers fail. Its scalability is evident as

Kafka can handle trillions of events per day, scaling horizontally by adding more brokers to the cluster. Low latency is maintained through efficient data handling and streamlined processing, allowing for real-time data pipelines that support time-sensitive applications. Moreover, Kafka's support for stream processing frameworks like Apache Flink and Kafka Streams enables the development of real-time analytics and complex event processing, adding further value to the data architecture.

Apache Kafka Features:

- Replication: Kafka's data replication feature ensures that data is copied across multiple brokers, providing redundancy and fault tolerance.

- Partitioning: Data partitioning enables parallel processing and enhances throughput by distributing the data load across multiple brokers.

- Integration: Kafka seamlessly integrates with various big data and analytics tools such as Hadoop, Spark, and ELK Stack, facilitating comprehensive data processing and analysis pipelines.

By elaborating on these components, we underscore the robustness and efficiency of the proposed architecture, highlighting its capability to handle real-time data processing, ensure data integrity, and provide scalable solutions for large-scale data management. And rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

D. Data Consumers

1.Managed Data:

A. Business Intelligence (Metabase):

Metabase is an open-source business intelligence tool that makes it easy for non-technical users to interact with data. It provides an intuitive, user-friendly interface that allows users to create queries and visualizations without needing to write SQL code. This democratizes data access across the organization, enabling business users to explore data, generate insights, and make data-driven decisions independently. Metabase supports a wide variety of visualizations, including charts, graphs, and dashboards, which can be customized and shared across teams. Additionally, it offers features such as scheduling automated reports and alerts, embedding analytics into applications, and integrating with various databases, further enhancing its usability and reach within the organization.

Metabase Features:

- Ease of Use: Metabase's straightforward setup and user-friendly interface reduce the learning curve, enabling quick adoption across the organization.

- Data Exploration: Users can drill down into data, create interactive filters, and visualize trends without needing technical expertise.

- Integration: Metabase connects to a wide range of databases, allowing seamless data integration from various sources.

C. *Analytical Applications (Apache Spark):*

Apache Spark is a powerful unified analytics engine designed for big data processing. It supports multiple programming languages (Java, Scala, Python, and R) and provides a comprehensive suite of libraries for various data processing needs, including streaming, SQL, machine learning, and graph processing. Spark's in-memory processing capabilities significantly accelerate data computation tasks, making it ideal for handling large datasets. Its structured APIs and high-level abstractions simplify complex data workflows, allowing data scientists and engineers to build sophisticated analytics applications. Moreover, Spark seamlessly integrates with Hadoop and other big data tools, enhancing its utility in a broader data ecosystem. Its ability to process data in real-time or batch mode makes it versatile for various analytical applications, from real-time analytics to large-scale data transformations and machine learning model training.

Apache Spark Features:

- Speed: Spark's in-memory processing capabilities significantly reduce the time required for data processing tasks.

- Versatility: Spark's support for different data processing paradigms (batch, streaming, ML, and graph) makes it a one-stop solution for diverse data analytics needs.

- Scalability: Spark can scale horizontally by adding more nodes to the cluster, accommodating growing data volumes and processing demands.

2.Self-Service Data:

A.Data Wrangling (Pandas - Python):

Pandas is a widely-used open-source data manipulation and analysis library for Python. It offers robust data structures, such as Data Frames, which allow for efficient handling of structured data. With Pandas, users can perform a wide range of data wrangling tasks, including cleaning, transformation, aggregation, and visualization. Its extensive set of functions and methods makes it a staple in the data science community for preprocessing data before analysis or feeding it into machine learning models. Pandas integrates well with other data science libraries like NumPy, SciPy, and Matplotlib, enhancing its capabilities for comprehensive data analysis workflows. The simplicity and flexibility of Pandas make it an indispensable tool for data analysts and scientists, enabling them to convert raw data into meaningful insights effectively

Pandas Features:

- Data Handling: Pandas excels at handling large datasets, offering efficient operations for data loading, cleaning, and transformation.

- Functionality: It provides a comprehensive set of data manipulation functions, making complex data wrangling tasks straightforward.

- Integration: Pandas integrates seamlessly with other Python libraries, enabling sophisticated data analysis workflows.

B.Ad-Hoc Query (Superset):

Apache Superset is an open-source data exploration and visualization platform designed to enable users to create interactive and informative dashboards easily. Superset provides a rich set of visualization options, including charts, maps, and tables, which can be customized to suit various data exploration needs. Its drag-and-drop interface allows users to build complex queries and visualizations without requiring

deep technical knowledge, making it accessible to a wide range of users within an organization. Superset also supports SQL queries for advanced users, offering flexibility in data exploration. The platform integrates with a multitude of data sources, including SQL-based databases and cloud-based data warehouses, making it a versatile tool for ad-hoc analysis. Users can create and share dashboards, facilitating collaborative data analysis and decision-making processes across teams.

Superset Features:

- Visualization Variety: Superset offers a wide range of visualization types, catering to different data exploration needs.

- Ease of Use: Its intuitive interface and drag-and-drop capabilities make it accessible for users without technical backgrounds.

- Collaboration: Superset facilitates collaborative data analysis by allowing users to create, share, and embed dashboards within the organization.

3.Harmonized Data Stores

A. Operational Applications (Docker):

Docker is an open-source platform that automates the deployment and management of applications within lightweight, portable containers. These containers encapsulate the application and its dependencies, ensuring consistent performance across different environments—from a developer's local machine to production servers. Docker containers are isolated from each other, enhancing security and reducing conflicts between applications. Additionally, Docker's orchestration tools, such as Docker Compose and Docker Swarm, allow for the efficient scaling and management of multi-container applications. By using Docker, organizations can achieve greater agility and flexibility in their development and deployment processes, reducing downtime and accelerating time-to-market for new features and services.

B.Analytical Data Stores (Apache Hive):

Apache Hive is a data warehousing solution built on top of the Hadoop ecosystem. It provides an SQL-like interface called HiveQL, enabling users to query and analyse large datasets stored in the Hadoop Distributed File System (HDFS). Hive's scalability makes it suitable for managing vast amounts of data, facilitating complex queries, and performing ETL (Extract, Transform, Load) operations. Its integration with Hadoop allows for distributed storage and parallel processing of data, enhancing performance and efficiency. Hive also supports various storage formats, such as ORC and Parquet, which optimize storage and query performance. By leveraging Hive, organizations can efficiently process and analyze large volumes of data, transforming raw data into valuable insights.

C. Business Intelligence Data Stores (Apache Druid):

Apache Druid is a high-performance, column-oriented, distributed data store designed for real-time data ingestion and fast analytical queries. It excels at providing low-latency query responses and high throughput for ingesting event data. Druid's architecture includes a combination of deep storage for long-term data retention and memory-optimized real-time nodes for recent data, ensuring both historical and real-time data can be queried seamlessly. Druid's indexing and partitioning strategies allow for efficient data organization and retrieval, making it suitable for Online Analytical Processing (OLAP) workloads. With Druid, organizations can perform interactive, ad-hoc queries on large datasets, enabling rapid data exploration and timely decision-making. Its capability to handle high query concurrency and provide high uptime makes it an ideal choice for business intelligence and analytical applications.

III. ARCHITECTURE

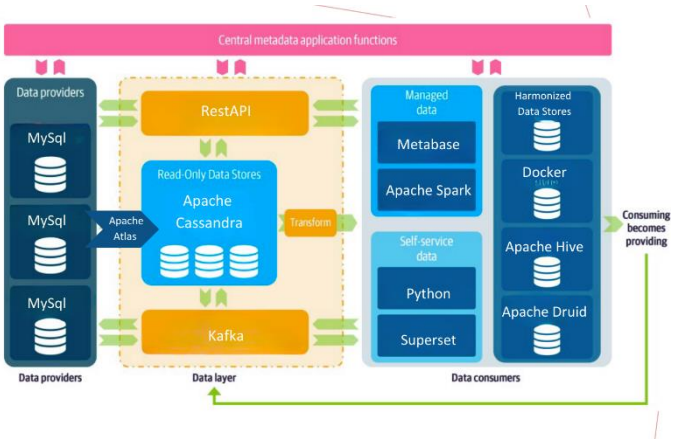

Figure-1

Figure-1 is the architecture of all the open source service providers. It is a robust and cost efficient data architecture for any of the organizations. The architecture employs MySQL for its reliable and consistent data storage capabilities. MySQL's ACID (Atomicity, Consistency, Isolation, Durability) compliance ensures that transactions are processed reliably, maintaining data integrity across different operations. On the other hand, Apache Cassandra is utilized for its high-performance search and read capabilities, designed to handle large volumes of data across distributed servers. Cassandra's decentralized architecture allows for continuous availability and high fault tolerance, making it ideal for applications requiring constant uptime and rapid data retrieval. By combining MySQL and Apache Cassandra, the architecture ensures both data consistency and fast, efficient access to data, catering to the needs of various applications within the system.

Apache Kafka plays a crucial role in managing real-time data streaming and messaging within the architecture. Kafka's high-throughput, low-latency platform can handle millions of events per second, making it perfect for real-time analytics and monitoring applications. The seamless data flow facilitated by Kafka ensures that data is processed in real-time, enabling timely event handling and decision-making. Kafka's ability to integrate with other components in the architecture, such as Cassandra and Spark, further enhances the system's capability to process and analyze data as it is generated, providing a comprehensive solution for real-time data management.

IV.ACRNONYMS AND ABBREVATIONS

API: Application Programming Interface
REST Representational State Transfer
SQL Structured Query Language
OLAP: Online Analytical Processing
ETL: Extract, Transform, Load
HDFS: Hadoop Distributed File System

CSV: Comma-Separated Values
BI: Business Intelligence
ML: Machine Learning
JSON: JavaScript Object Notation
HTTP: Hypertext Transfer Protocol
JWT: JSON Web Token
ORM: Object-Relational Mapping
IOT: Internet of Things
ELK: Elasticsearch, Logstash, Kibana

V.TABLE

Table-1

Component	Functionality	Key Features
MySQL	Reliable and consistent data storage	High availability, strong ACID compliance
Apache Cassandra	High-performance search and read capabilities	Scalability, high availability, multi-data center replication
Kafka	Real-time data streaming and messaging	Durability, scalability, low latency
Metabase	Business intelligence and data visualization	User-friendly, wide range of visualizations
spark	Data processing and analytics	In-memory processing, support for multiple languages
jupyter	Interactive data exploration and visualization	Interactive, supports various programming languages
hive	Data warehousing and SQL-like queries on large datasets	Integration with Hadoop, support for various storage formats

Table-1 helps understand the functionality and key features of the open-source components in architecture

IV. CONCLUSION

The proposed data architecture effectively integrates multiple open-source tools to deliver a comprehensive solution for real-time data processing, storage, and analytics. Each component in the architecture plays a specific role, contributing to the overall system's efficiency, scalability, and reliability. The inclusion of Apache Atlas for master data management is a key enhancement, providing robust data governance and metadata management. Atlas ensures that all data assets are properly documented, classified, and governed, enhancing the architecture's reliability and compliance with data management standards. By leveraging open-source technologies, the architecture offers a cost-effective and flexible solution that can be tailored to meet the unique needs of different organizations. The real-time processing capabilities provided by Apache Kafka, combined with the data storage and retrieval strengths of MySQL and Apache Cassandra, ensure that data is both consistent and readily accessible. The analytical tools—Metabase, Apache Spark, and Jupyter Notebooks—provide powerful platforms for data analysis and visualization, enabling users to gain insights and drive business decisions. Furthermore, Docker's containerization capabilities ensure efficient application deployment and scaling, while Apache Hive and Apache Druid offer robust solutions for managing and querying large datasets.

Overall, this architecture is well-suited for organizations seeking to harness the power of open-source technologies for comprehensive data management and analytics. It provides a scalable, reliable, and flexible framework that supports real-time data processing, robust data governance, and insightful data analysis, positioning organizations to effectively manage and leverage their data assets.

VII.ACKNOWLEDGMENT

I would like to express my profound gratitude to Professor Geoffrey Thomas for his invaluable guidance and expert advice throughout the development of this paper. Thank you, Professor Thomas, for your unwavering support and encouragement.

VIII. REFERENCES

[1] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A Distributed Messaging System for Log Processing. LinkedIn.

[2] Lakshman, A., & Malik, P. (2010). Cassandra: A Decentralized Structured Storage System. ACM SIGOPS Operating Systems Review, 44(2), 35-40.

[3] Emre Akin, "What is Kafka?" Medium, October 2023. [https://medium.com/@cobch7/what-is-kafka 9cc8591d2063]