Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 6): pp. 6468-6475
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.06.631

Data Lake Ingestion Tools: A Comprehensive Study

Mrs. Asmita P. Dongaonkar VMV Comm, JMT Arts and JJP Sci College Nagpur, Nagpur, Maharastra, India. asmita.dogaokar@gmail.com

Dr. Vaibhav R. Bhedi VMV Comm, JMT Arts and JJP Sci College Nagpur, Nagpur, Maharastra, India vaibhav_bhedi@rediffmail.com

Abstract
In recent years, Big Data applications have become increasingly vital for organizations and industries aiming to achieve a competitive advantage and discover new trends and insights. A crucial initial step in Data Lake is data ingestion. This paper offers a thorough review of some of the most widely used methods for Data Lake ingestion and ensures that data is accurately classified, securely managed, and easily accessible for analysis, thus enabling the utilization of diverse datasets to generate insights and support decision-making processes.

Keywords: big data, data ingestion, data lake, DL

Introduction

Due to the significant increase in data generated by human activities, organizations have often spontaneously created Data Lakes. These Data Lakes are formed by the physical grouping of datasets related to the same activity, allowing for centralized storage and access to large volumes of diverse data. A Data Lake is a vast collection of data that includes structured, semi-structured, and unstructured datasets. These datasets typically have the following characteristics: (1) they can be stored on heterogeneous systems, they can be used independently of one another, (3) some may contain raw data, meaning the data is stored in its original form without being organized for specific uses, and (4) the types and formats of the data can vary widely. In practice, a Data Lake can include a variety of datasets, such as relational databases, object databases, Comma Separated Values (CSV) files, text files, and spreadsheet folders (Alwidian, J., Rahman, S. A., Gnaim, M., & Al-Taharwah, F. (2020)). The large volume of data in a Data Lake serves as a crucial source of knowledge for business decision-makers. This data can be organized according to a multidimensional data model to support certain types of decision processing. However, the heterogeneity of storage systems and the diversity of content in the Data Lake pose significant challenges for data utilization in decision-making. Ingestion is the process of extracting data from various sources and transferring it to a repository, where it can then be transformed and analyzed. For example, large volumes of data from various sources are ingested into a Data Warehouse and utilized in the context of information retrieval on the Web.

To bring data into Data Lakes, tools like Flume, NiFi, and Kafka are commonly utilized. For a Data Lake to act as a central repository for all company data, integrating multiple ingestion tools is often essential. In this paper, we examine the process of ingesting data into a large Data Lake for the purpose of storing the company's own data. We explore the challenges associated with data ingestion into the Data Lake, including data integration complexities, data quality issues, scalability concerns, and ensuring data consistency.

What is data Ingestion?

A data ingestion framework captures data from multiple sources and ingests it into a big data lake. It securely connects to different sources, captures changes, and replicates them in the data lake, ensuring the data lake remains consistent with the source systems and serves as a single repository of enterprise data. A standard ingestion framework consists of two main components: the Data Collector and the Data Integrator. The Data Collector is responsible for gathering or pulling data from various sources, while the Data Integrator handles ingesting the collected data into the data lake. The design and implementation of these components can be adapted based on the specific big data technology stack being used. Data ingestion ensures that data from multiple sources is collected, consolidated, and made available for analysis, reporting, or other uses.

Data ingestion process should be handle the different volume, speed and variety of data. It can be batch data ingestion or Stream data ingestion. This paper discussed the Big Data ingestion process with different tools for batch and stream ingestion such as Sqoop, NIFI, Flume and Kafka. Each tool is discussed with its' features, architecture and real use case. It has a comparison for big data ingestion tools based in different criteria, this comparison will help users to choose the tool that satisfies their needs. Also, it mentioned the data preparation process that aims to clean, validate and reduce the ingested data.it mentioned some tools for data preparation like Hive, Impala, Storm and Spark.

In the context of big data projects, the landscape of data sources is vast and constantly evolving. The sheer volume of data involved, as well as the diversity in its sources and formats, presents both challenges and opportunities for organizations. Data sources can broadly be classified into two main categories: internal and external. (Bucur, C., 2015) (Erraissi, A., Belangour, A., &Tragha, A., 2018)

Internal Data Sources

Internal data sources are those that an organization controls directly. This data is typically collected from the day-to-day operations of the company and is stored within its internal systems. It often consists of structured data, meaning it is organized in a predefined manner, usually in rows and columns within databases. Structured data is relatively easy to manage and analyze due to its consistent format. Examples of internal data include:

1. **Transactional Data:** This includes data generated from business transactions, such as sales records, purchase orders, and billing information. This data is often stored in relational databases and can be analyzed to understand sales trends, financial performance, and customer purchasing patterns.

2. **Customer Data:** Information about customers, such as contact details, demographics, purchase history, and preferences. This data can be used for targeted marketing, customer service, and loyalty programs.
3. **Operational Data:** Data related to the internal operations of an organization, such as inventory levels, supply chain logistics, and employee records. This data helps in optimizing internal processes and improving efficiency.

External Data Sources

External data sources originate from outside the organization and are not under its direct control. These sources are highly diverse and can include structured, semi-structured, and unstructured data. The analysis of external data can provide valuable insights that complement the internal data, offering a more comprehensive view of the market, customers, and competitors. Key external data sources include:

1. **Social Media:** Platforms like Twitter, Facebook, Instagram, and LinkedIn generate a tremendous amount of data daily. This includes posts, comments, likes, shares, tweets, and profile information. Social media data is rich in insights about public opinion, brand sentiment, customer feedback, and trends. For instance, analyzing social media conversations about a new product can reveal how well it is being received by the public and highlight areas for improvement.
2. **Log Files:** Websites and online applications generate log files that record user interactions, such as page views, clicks, and navigation paths. This data can be analyzed to understand user behavior, preferences, and engagement patterns, which can inform website design, content strategy, and marketing efforts.
3. **Sensors and Machines:** The Internet of Things (IoT) has led to a proliferation of data from sensors and machines. This includes data from smart devices, medical equipment, industrial machinery, and environmental sensors. For example, smart meters in homes collect data on energy usage, which can be used for optimizing energy consumption and detecting anomalies. Similarly, medical devices can monitor patients' vital signs in real-time, providing critical data for healthcare providers.
4. **Geospatial Data:** This type of data is collected from GPS-enabled devices, mobile phones, and other location-based services. It includes information about geographic locations, movements, and patterns. Geospatial data can be used in various applications, such as mapping services, location-based marketing, and urban planning.

Types of Data

In big data projects, the data collected can be categorized into three main types based on its structure: (Erraissi, A., Belangour, A., &Tragha, A., 2018)

1. **Structured Data:** This is data that is highly organized and easily searchable. It is stored in a tabular format with rows and columns, and each piece of information has a defined relationship with the others. Examples include spreadsheets,

databases, and tables in relational databases. Structured data is easy to enter, store, query, and analyze, making it a valuable asset for organizations.

2. **Unstructured Data:** Unlike structured data, unstructured data does not follow a specific format or organizational structure. This makes it more challenging to analyze and process. Examples of unstructured data include emails, social media posts, videos, audio files, and images. Despite its complexity, unstructured data often contains rich insights, especially when analyzed using advanced techniques like natural language processing (NLP) and machine learning.
3. **Semi-Structured Data:** Semi-structured data lies between structured and unstructured data. It does not conform to a rigid structure, but it does contain tags or markers that separate different elements and enforce hierarchies. XML and JSON files are common examples of semi-structured data. This type of data is more flexible than structured data and can accommodate a wider range of information, making it useful for representing complex data relationships.

Overall, big data projects rely on a mix of these data types and sources to extract valuable insights. The ability to effectively manage, integrate, and analyze this data is crucial for organizations to make informed decisions, optimize operations, and stay competitive in an increasingly data-driven world.

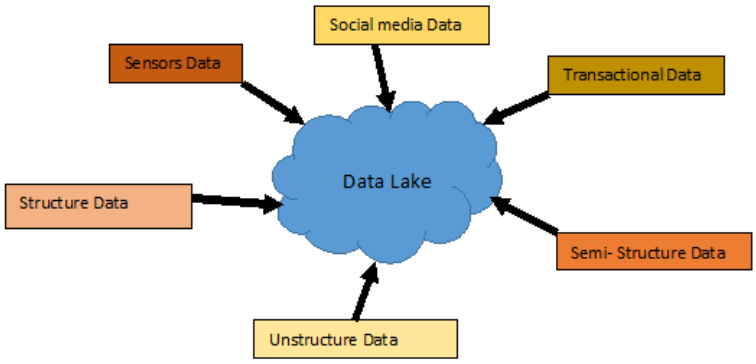

Fig. 1 Source of Data lake

To effectively complete the data ingestion process, it is crucial to select the right data ingestion tool that aligns with the specific business case. Several key parameters must be considered to choose the most suitable tool, as they directly impact the efficiency and effectiveness of the data ingestion process.

Key Parameters in Data Ingestion

1. **Data Size:** This parameter refers to the volume of data generated by various sources that need to be ingested. The tool must be capable of handling large datasets efficiently, whether the data comes from internal systems, social media, sensors, or any other source. The capacity of the tool to scale and manage large amounts of data is essential to ensure smooth ingestion without bottlenecks.

2. **Data Format:** Data can come in different formats, which may include:
    - **Structured Data:** Organized in a tabular format, such as databases or spreadsheets.
    - **Unstructured Data:** Lacks a predefined structure, including images, videos, text documents, and social media posts.
    - **Semi-Structured Data:** Contains elements of both structured and unstructured data, like XML or JSON files.

    The chosen data ingestion tool must be versatile enough to handle these diverse data formats, enabling seamless integration and processing.
3. **Data Frequency:** This parameter defines how often data is ingested, which can be either in real-time or in batches:
    - **Real-Time Ingestion:** Data is processed as soon as it is received. This is crucial for applications that require immediate data analysis and action, such as fraud detection, real-time analytics, or monitoring systems.
    - **Batch Ingestion:** Data is collected and stored in batches and then ingested at specific intervals. This approach is suitable for scenarios where real-time processing is not necessary, and data can be processed periodically.

    Understanding the frequency requirements of the business use case is vital in selecting a tool that can support the appropriate mode of ingestion.

4. **Data Velocity:** This refers to the speed at which data flows from various sources into the system. Data velocity can vary significantly depending on the nature of the business and the sources of data. For instance, a news website may experience spikes in data traffic during major events, while a retail website may see increased data flow during sales periods. The data ingestion tool should be capable of handling varying data velocities, ensuring consistent and reliable data ingestion even during periods of high data traffic.

Study of different Data Ingestion Tool

Selecting the right data ingestion tool requires a comprehensive evaluation of these parameters. The tool must be compatible with the expected data size, format, frequency, and velocity to meet the specific needs of the business. Additionally, factors such as ease of use, scalability, integration capabilities with existing systems, and cost-effectiveness should also be considered. By carefully assessing these factors, organizations can ensure a robust and efficient data ingestion process that supports their data-driven decision-making and operational needs.

Sqoop Apache

Sqoop is a tool designed for efficiently transferring large amounts of data between Hadoop and relational databases or mainframes. It facilitates the import of data from various relational database management systems (RDBMS) like Oracle, MySQL, or

mainframe systems into the Hadoop Distributed File System (HDFS). Once the data is in HDFS, it can be processed using Hadoop'sMapReduce framework, and the processed data can then be exported back to an RDBMS.

The connection between Sqoop and the RDBMS is established using a JDBC connector, which relies on the schema of the RDBMS for the imported data. Sqoop leverages MapReduce's parallel processing capabilities to efficiently handle large datasets. As a result, the output of a Sqoop import process is typically multiple files stored in HDFS. These files can be in different formats, such as delimited text files, binary Avro files, or Sequence Files, depending on the specific needs and configuration(Alwidian, J., Rahman, S. A., Gnaim, M., & Al-Taharwah, F. (2020)).

Sqoop's ability to integrate with RDBMS and Hadoop ecosystems makes it a powerful tool for data warehousing, ETL (Extract, Transform, Load) processes, and data analytics. It is commonly used to analyze data from relational databases, apply machine learning algorithms, or aggregate data in Hadoop, and then export the results back to a relational database for further use.

Figure 2. Sqoop Functionality (Cheng, Y., Zhang, Q., & Ye, Z., 2019)

NIFI Apache

NIFI is a dataflow system that can collect, transform, process and route data.it was built on flow-based programming concept, it was designed to automate and manage the flow of data between systems. (Peng. R.. 2019)

NIFI is Java based and executed within JVM on a host operating system, as shown on figure 3 below the architecture of NIFI consist of different components, Web Server which is responsible about hosting NiFi's HTTP-based command and to enable the user to access NIFI via web based interface. Flow Controller which is responsible about providing and scheduling threads for execution. FlowFile Repository which is the area where NIFI track the status updates about the flowfiles. Content Repository that holds the content of flowfiles and Provenance Repository that holds provenance event data.

NIFI is able to run within a cluster, each node in the NIFI cluster complete the same tasks but interact with different set of data. The cluster is managed by cluster coordinator which is elected by ache Zookeeper.(Isah, H., &Zulkernine, F. (2018)).

NIFI has friendly web based user interface that allow users to drag and drop components to build the dataflow, the components can be started and stopped in real time also the errors and statistics can be viewed easily NIFI buffering all the queued data and allow setting prioritization schemes to indicate how the data will be retrieved from the queue. NIFI provide data provenance module in order to track the data from the start of the flow

until the end. The implemented dataflow is secure since NIFI use secured protocols like SSL, HTTPS, SSH and other encryptions.
A processor is an atomic element in NiFi dataflow which can do different tasks, it can read data from multiple resources, route, transform and publish data to external resources. For batch data ingestion NIFI processors can read the data from different sources, it can be any SQL database server like Oracle and MySQL, or NoSQL databases like MongoDB, or pulling data with different format from local or remote systems.

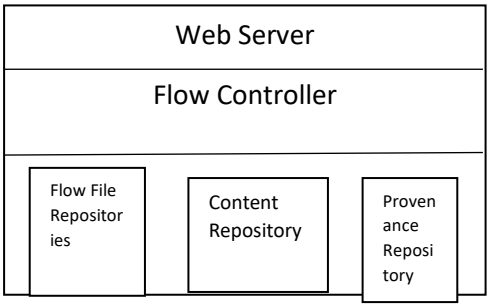

Fig3: NIFI Architecture

Flume Apache
Flume is a distributed, reliable, and efficient service designed for the real-time ingestion, collection, aggregation, and transportation of large volumes of data, particularly suitable for online analytical applications. It ensures robust fault tolerance and data consistency, making it a vital tool for real-time data analytics, refining, and visualization. Flume operates using a pipeline-like architecture, where data flows from sources (which receive data from generators) to channels (which act as bridges), and finally to sinks (which deliver data to destinations such as HDFS). Its high-performance scalability and integration capabilities with systems like Hadoop make it an essential component for managing dynamic and large-scale data flows, as demonstrated in various practical applications such as sensor data collection and processing in smart systems.

Flume has been utilized in various research and practical applications.
Flume's robust architecture and features make it an essential tool for real-time data ingestion and processing, particularly in environments that require high reliability, scalability, and efficient data flow management. It is widely used in scenarios involving large-scale data collection from sensor networks, log aggregation, and real-time analytics.

Kafka Apache
Kafka is a distributed streaming tool that provides a unified, high-performance data feed and messaging brokering system. Its most notable feature is low latency, as all processing occurs in memory to avoid the access latency associated with hard disks. Kafka is designed to handle large volumes of messages with low latency and high fault tolerance. The system comprises three major components: the broker, consumer, and producer. The

broker acts as the server and is responsible for ensuring fault tolerance, a key feature of Kafka. Producers send messages to consumers via the broker, which serves as a channel to route and differentiate messages. Kafka's architecture includes multiple nodes and brokers, creating a high-performance, real-time data channel.

Kafka operates with two main processes: distributing and publishing messages. Its architecture supports clusters of servers, each capable of handling thousands of clients for extensive read and write operations. These clusters act as a central point for data in large organizations, maintaining logs of messages with sequential IDs to ensure fault tolerance. Kafka's scalability allows the system to expand elastically without interruption, distributing data across machines even if individual machine capacity is limited. Kafka's robust design, high scalability, and message consistency make it a superior choice for real-time data ingestion and analytics compared to other tools like Flume. It is used widely in various applications, including ingesting social media data for analytics, managing large-scale data flows in enterprises, and supporting real-time data processing in diverse environments.

Conclusion

The increasing diversity of data formats and the massive volume of generated data make the ingestion and preparation stages critical in any Data Lake. A thorough understanding of these processes and the appropriate tools to use is essential for efficient data management and processing. By reviewing various tools and their characteristics, this paper provides valuable comparative study, ultimately aiding in the successful implementation of Data Lake ingestion.

References:

Alwidian, J., Rahman, S. A., Gnaim, M., & Al-Taharwah, F. (2020). Big data ingestion and preparation tools. Modern Applied Science, 14(9), 12-27.

Bucur, C. (2015, July). Using big data for intelligent businesses. In Proceedings of the Scientific Conference AFASES (Vol. 2, pp. 605-612).

Erraissi, A., Belangour, A., &Tragha, A. (2018). Meta-Modeling of Data Sources and Ingestion Big Data Layers. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3185342

Peng, R. (2019). Kylo Data Lakes Configuration deployed in Public Cloud environments in Single Node Mode. DiVA, id: diva2:1367021

Isah, H., &Zulkernine, F. (2018). A Scalable and Robust Framework for Data Stream Ingestion. 2018 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/BigData.2018.8622360