International Journal of Advanced Scientific Research in Engineering Trends and Technology

Volume 1 Issue 1,NOV-DEC 2025
www.ijengt.com

Design and Implementation of an Open-Source ETL Framework on AWS Cloud using Big Data Tools

C. Gahane[1], D.Triupathi Rao[2]
[1]Research Scholar,[2]Assistant Professor
Electronics and Telecommunication, Priyadarshani Collegeof Engineering,Nagpur,Maharashtra,India.

Abstract:

The fast rise of data volume, diversity, and velocity has posed substantial challenges to traditional Extract, Transform, and Load (ETL) approaches, prompting the development of new tools and technologies. This research study investigates the usage of open-source ETL frameworks in conjunction with big data tools and cloud services, with a focus on low-cost solutions that maintain scalability and dependability. Commercially licensed ETL tools are typically prohibitively expensive, making them inaccessible to smaller enterprises. In contrast, open-source alternatives offer solutions that are flexible, inexpensive, and adaptable. This study critically explores how open-source ETL frameworks can be used on cloud platforms like Amazon Web Services (AWS) to create resilient, scalable, and efficient data processing pipelines. It compares several open-source ETL tools, such as Talend, Pentaho Data Integration, Kettle, and Apache NiFi. The paper also addresses how cloud-based orchestration tools such as Apache Airflow and AWS Elastic MapReduce (EMR) can be used to achieve cost-effective, resilient data processing.

Introduction

Data management is growing more challenging as data volume and variety increase exponentially. Traditional ETL methods are straining to keep up, prompting the development of more complex solutions that can handle large amounts of data [1]. Big data tools are intended to manage large-scale data processing efficiently, and open-source ETL frameworks provide a low-cost alternative to commercial ETL systems. These frameworks may be implemented in scalable, flexible environments using cloud services such as AWS, making them a viable option for modern organizations.

The primary research question posed in this study is: "How can an open-source ETL framework for big data be developed using cloud services to be both cost-effective and feature-rich while incorporating the necessary resilience provided by cloud platforms?" This research begins with an overview of the traditional ETL ecosystem, then moves on to open-source ETL frameworks and big data tools, and then compares important frameworks. Finally, the article investigates how cloud orchestration services such as AWS might improve the capabilities of these open-source tools, making them more appropriate for modern big data concerns.

Traditional ETL Challenges in the Big Data Era

The introduction of Big Data has significantly altered the manner in which organizations manage and analyze information [2]. Nevertheless, it has also introduced new obstacles for conventional Extract, Transform, and Load (ETL) processes. These challenges are a result of the necessity for real-time analytics and data-driven decision-making, as well as the overwhelming volume, velocity, and variety of Big Data [3].

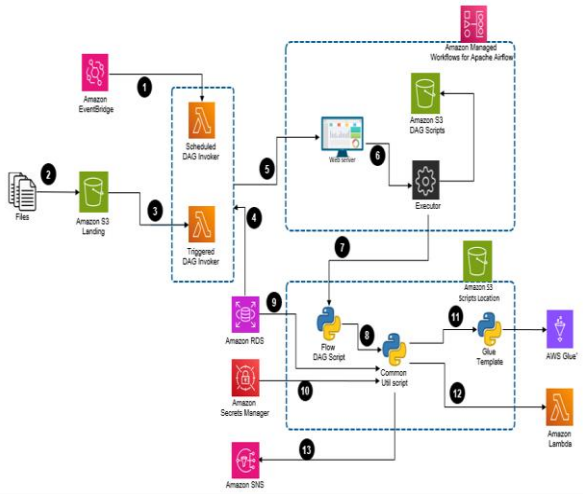

Figure 1.1: Reusable ETL framework architecture

Volume

- Massive Datasets: Big Data frequently involves datasets that are too large to be processed by conventional ETL tools. This can result in heightened processing times and performance constraints [4].
- Storage Challenges: The storage and management of large databases necessitate the implementation of specialized storage solutions and the application of efficient data compression techniques [5].

Acceleration

- Real-Time Processing: In order to facilitate time-sensitive decisions, numerous Big Data applications necessitate near-real-time processing. The velocity of incoming data may be difficult for traditional ETL tools to keep up with, as they are frequently batch-oriented [6].
- Data Streams: ETL processes encounter distinctive obstacles when managing continuous data streams, such as those produced by IoT devices or social media platforms [7].

Diverseness

- Big Data comprises a diverse array of data types, such as structured data (e.g., relational databases), semi-structured data (e.g., JSON, XML), and unstructured data (e.g., text, images, audio). These diverse data formats necessitate ETL tools that are capable of accommodating them [8].
- Data Quality: It is essential to guarantee data quality in Big Data environments to ensure precise analysis. This can be difficult because of the potential for errors or inconsistencies and the diversity of data sources [9].

Intricacy

- Complex Transformations: Big Data applications frequently necessitate intricate data transformations, including aggregations, joins, and machine learning algorithms. These intricate operations may not be proficiently managed by conventional ETL tools [10].
- Data Governance: The volume and diversity of data in Big Data

environments can make data governance and compliance management a complex task.

Expandability

- Horizontal Scalability: In order to accommodate the growing volume of data, ETL processes must be capable of scaling horizontally. In order to accomplish this, it is necessary to implement distributed processing frameworks and implement effective resource management [11].
- Vertical Scalability: In certain instances, ETL processes may also require vertical scaling through the utilization of specialized accelerators or more powerful hardware [12].

Cost

- Infrastructure expenses: Big Data storage and processing can be costly, particularly for organizations that require the acquisition of specialized hardware or cloud-based solutions.
- Costs of Talent: It can be difficult to identify and retain data architects and analysts who are capable of effectively managing Big Data ETL processes.

In order to confront these obstacles, organizations are employing contemporary data processing frameworks such as Apache Spark, which are specifically engineered to manage large-scale data sets and facilitate real-time processing [13]. Furthermore, cloud-based data warehousing and analytics platforms provide cost-effective and scalable solutions for Big Data ETL.

Traditional ETL systems were designed to handle structured data from a small number of sources, usually relational databases. However, the modern data landscape is even more complex, with data arriving in a variety of formats from a wide range of sources, including social media, IoT devices, and corporate systems [14]. Traditional ETL methods face several issues when dealing with massive data, including:

- Scalability: Traditional ETL solutions struggle with the massive amount of data that needs to be processed.
- Data Variety: Traditional ETL technologies struggle to extract and transform data that is structured, semi-structured, or unstructured.
- Latency: Big data frequently requires real-time processing, which many standard ETL methods cannot handle.
- Cost: Commercial ETL tools have substantial licensing fees, making them prohibitively expensive for many organizations.

Emergence of Big Data Technologies

Big data technologies, such as Hadoop, Spark, and Kafka, have emerged to address the issues of modern data management [15]. These systems are distributed, scalable, and can handle data in real time. They constitute the foundation of modern ETL procedures, enabling for the effective extraction, transformation, and loading of data from many sources at scale.

Apache Hadoop

Apache Hadoop is a popular big data platform, well recognized for its distributed storage and processing capabilities. Hadoop's Hadoop Distributed File System (HDFS) can store enormous datasets, and

its MapReduce engine can process data in parallel across numerous nodes.

Apache Spark

Apache Spark was created for quick, in-memory data processing. It works well with iterative algorithms and interactive applications, making it an effective tool for real-time data processing. Spark's ability to retain data in memory dramatically reduces latency in data processing.

Kafka & Airflow

Kafka is a high-throughput, distributed messaging system capable of real-time data input and processing. It is frequently utilized in scenarios requiring real-time analytics. Airflow, on the other hand, is a workflow orchestration tool that facilitates the management and scheduling of complicated workflows, particularly in ETL procedures involving large amounts of data.

Open-Source ETL Frameworks

Because of their affordability, scalability, and open-source nature, a number of ETL frameworks have gained popularity. These frameworks provide a wide range of features that can compete with commercially available ETL tools.

Talend

Talend is a Java-based open-source ETL solution that supports a variety of data sources and has a drag-and-drop interface for creating ETL workflows. Talend provides both a free community edition and a premium enterprise edition, making it suitable for organizations of all sizes. It supports a variety of big data technologies, including Hadoop and Spark, and is compatible with cloud platforms such as AWS.

Pentaho Data Integration (PDI

Pentaho Data Integration (PDI), formerly known as Kettle, is a powerful ETL solution with extensive functionality for data transformation and integration. PDI is similarly Java-based and can handle a wide range of data sources. Its drag-and-drop interface makes it easier to create ETL pipelines, and there is a free community edition available.

Apache NiFi

Apache NiFi is a strong data integration solution that automates the movement of data between systems. It is especially handy for real-time data intake and can handle a variety of protocols and data formats. NiFi's flow-based programming architecture makes it simple to create ETL workflows, and the community edition is free.

Comparative Analysis of Open-Source ETL Tools

There are benefits and drawbacks to each of these open-source ETL solutions. Talend's drag-and-drop interface and extensive list of supported data sources make it especially user-friendly. PDI has significant data transformation capabilities, yet it might be complicated for novices. NiFi excels in real-time data processing but may necessitate more technical knowledge for setup and configuration.

Cloud Services for Open-Source ETL Frameworks

Cloud platforms such as AWS are a perfect setting for installing ETL frameworks since they provide scalable infrastructure, distributed computing, and integrated big data services. The primary benefits of adopting cloud services for ETL procedures are:

- Scalability: Cloud platforms can scale horizontally, enabling the

processing of massive volumes of data.

- Cost Efficiency: Organizations can pay for cloud resources as they utilize them, lowering initial expenses.
- Many cloud platforms provide managed big data services, which make it easier to build and manage ETL procedures.

Amazon Elastic MapReduce (EMR)

AWS EMR is a cloud-native big data platform that streamlines the deployment of Hadoop, Spark, and other big data frameworks. It enables enterprises to handle massive amounts of data with a fully managed, scalable, and cost-effective solution. EMR integrates well with open-source ETL tools such as Talend and Apache NiFi, resulting in a powerful environment for massive data processing.

AWS Glue

AWS Glue is a fully managed ETL solution that allows you to easily prepare and load data for analytics. Glue automatically detects and catalogs data, making ETL procedures more efficient. While Glue is not an open-source technology, it can be combined with open-source ETL frameworks to improve their functionality.

Apache Airflow for AWS

Apache Airflow can be used on AWS to manage complex ETL procedures. Organizations can create robust, scalable ETL pipelines for real-time data processing and big data workloads by connecting Airflow with AWS services such as EMR, S3, and Lambda.

Developing an Open-Source ETL Framework on AWS

An open-source ETL framework can be created by combining big data tools for parallel processing with cloud services for scalable infrastructure orchestration. The steps below outline how such a framework can be developed.

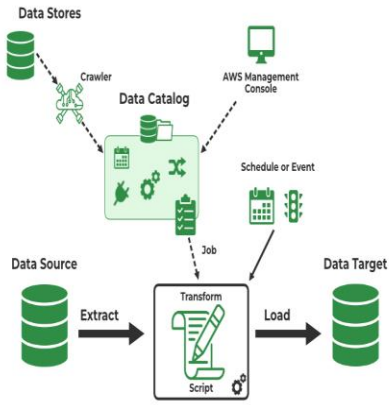

Figure.2: AWS Glue Architecture

1. Identifying requirements.

The first step is to identify the organization's specific data processing needs. This comprises determining data volume, data source variety, and processing delay requirements.

2. Choosing Big Data Tools.

The proper big data tools must be chosen in accordance with the defined needs. Kafka and NiFi are perfect for processing data in real-time, in contrast to Hadoop and Spark, which are commonly used for batch processing.

3. Choosing Cloud Services.

The next step is to determine which cloud services will host the ETL framework. AWS offers a variety of services for building scalable ETL pipelines, including EMR, Glue, S3, and Lambda.

4. Developing the ETL Framework

The ETL framework is created by combining the specified big data technologies with cloud services. Data can be ingested with Kafka, processed with Spark on EMR, and saved in S3. Airflow can be utilized to manage the entire workflow.

5. Testing and deployment.

Once the framework has been created, it should be rigorously tested with real-world datasets to ensure that it can manage the necessary data volumes and processing rates. Following testing, the framework can be deployed in a production environment on AWS.

Case Study: Creating a Big Data ETL Pipeline using Talend and AWS EMR

The design of scalable and cost-effective ETL (Extract, Transform, Load) pipelines has become imperative due to the growing data volumes, real-time data processing demands, and increasing complexity of data sources. This case study concentrates on the integration of Talend, an open-source ETL tool, with Amazon Web Services Elastic MapReduce (AWS EMR) to establish a large data ETL pipeline that is scalable, efficient, and robust. We investigate the potential of combining Talend and AWS EMR to enhance decision-making through real-time analytics, automate data processing, and manage large-scale data.

A global e-commerce company was encountering difficulties in administering its expanding data infrastructure, which involved the management of large datasets from a variety of sources, including social media platforms, customer databases, IoT devices, and sales channels. The current ETL system was constructed using conventional methods, which were unable to accommodate the growing volume and diversity of data. Consequently, they required a more scalable solution that could manage real-time and batch data processing with minimal disruption and cost-effectiveness.

Solution Development

Using Talend as the ETL tool and AWS EMR for data processing at scale, the solution was constructed around the concept of a distributed ETL pipeline. The ETL pipeline's design and implementation are delineated in the subsequent steps:

1. Data Ingestion

Data was obtained from numerous sources, including:

- Social Media Platforms: Real-time processing was necessary for data transfers from platforms such as Twitter and Facebook.
- Sensor data from IoT devices that perpetually monitor supply chain activities.
- Enterprise Databases: Data that is structured and derived from customer and sales databases.
- CSV/JSON Files: Daily receipt of batch data files in a variety of formats.

The data was extracted from these sources using Talend's native connectors. Talend's integration with Apache Kafka was implemented to facilitate real-time data ingestion for streaming data.

2. Talend for Data Transformation

The subsequent phase involved the transformation of the data that had been ingested. The following ETL tasks were designed using Talend's graphical interface:

- Data Cleansing: The process of eliminating duplicate records, managing lacking data, and standardizing data formats.
- Data Enrichment: The process of integrating basic data with external data sources to provide supplementary context. For instance,

utilizing IP addresses to geotag customer locations.

- Data Aggregation: The process of condensing extensive datasets to generate meaningful metrics for subsequent analysis.

The transformation logic was developed by utilizing Talend's pre-built components, including tMap (for mapping and transformation), tAggregateRow (for aggregation), and tFilterRow (for data filtration). The Talend Jobs that were exported from the ETL workflows were subsequently deployed on AWS EMR for distributed processing.

3. Data Transfer to AWS S3 and RDS

The cleaned and enriched data was transferred into two primary destinations following the transformation:

- Amazon S3: Functioned as the primary data lake for the storage of both unprocessed and processed data. The S3 storage enabled effortless access to data for downstream processing and analytics.
- Transformed data was imported into relational databases such as MySQL and PostgreSQL using Amazon RDS. From there, it was utilized for reporting and analysis using Business Intelligence (BI) tools such as Tableau.
- Talend's tS3Put and tMysqlOutput components were employed to automate the loading procedure for S3 uploads and RDS inserts, respectively.

4. AWS EMR Distributed Processing

In order to manage the substantial volume of data, Apache Spark tasks were executed on multiple nodes using AWS EMR. The auto-scaling feature of EMR dynamically adjusted the cluster's capacity in response to workload demand, thereby optimizing cost while preserving performance.

Talend's inherent integration with Apache Spark enabled the execution of Talend Jobs within Spark's distributed processing environment. This enabled the rapid processing of large datasets by parallelizing jobs across numerous nodes in the EMR cluster.

5. Automation and Workflow Orchestration

Apache Airflow, which was deployed on AWS, was employed to coordinate the entire ETL process. The scheduling, monitoring, and management of dependencies between various ETL tasks were facilitated by Airflow. Airflow DAGs (Directed Acyclic Graphs) initiated each Talend Job and additionally monitored the success or failure of the respective tasks.

In addition, AWS Lambda functions were employed to initiate specific ETL workflows in response to data events. For instance, when a new file was uploaded to S3, Lambda would initiate an Airflow DAG to process the file.

Apache Kafka Real-time Processing

The ETL infrastructure was integrated with Apache Kafka to facilitate real-time data ingestion and processing. Kafka facilitated the collection of streaming data from IoT sensors and social media platforms, which was subsequently processed in real-time by Talend jobs that were operating on AWS EMR. The results were immediately transmitted to downstream systems, which provided valuable insights for decision-making.

Monitoring and Error Management

Monitoring was an indispensable element of the ETL pipeline. The health of the EMR clusters and Talend Jobs was monitored using AWS CloudWatch. Alerts were transmitted to the operations team for immediate action in the event of any logging of errors or performance issues. Talend's error-handling components, such as tLogCatcher, were configured to capture and log errors at each stage of the pipeline.

Enhancing Performance

In order to guarantee optimal performance, numerous methodologies were implemented:

- Parallelism: The time consumed to process large datasets was considerably reduced by configuring Spark jobs to parallelize tasks across multiple nodes in the EMR cluster.
- Auto-scaling: EMR's auto-scaling capability guaranteed that cluster resources were automatically adjusted in accordance with the workload, thereby minimizing expenses during periods of minimal activity.
- Partitioning of Data: The data in Amazon S3 was partitioned by date to facilitate quicker retrieval and more efficient queries.

Compliance and Security

Security was a concern during the development of the pipeline:

- Data Encryption: The AWS Key Management Service (KMS) was employed to encrypt data stored in Amazon S3. In the same vein, SSL was implemented to encrypt data in transit.
- Access Control: Configured AWS Identity and Access Management (IAM) roles to regulate access to the EMR cluster and other AWS services. The data pipeline could only be accessed or modified by authorized personnel.
- Audit Logging: To guarantee industry compliance, all access and data manipulation activities were recorded using AWS CloudTrail.
- Outcomes

A highly scalable, cost-effective, and robust large data ETL pipeline was achieved through the integration of Talend and AWS EMR. The pipeline accomplished the following:

- Scalability: The pipeline was capable of processing terabytes of data on a daily basis, thereby satisfying the company's expanding data requirements.
- Real-time Processing: The pipeline was able to manage real-time data streams by utilizing Kafka, which allowed for faster decision-making based on the most recent information.
- Cost Savings: As opposed to conventional on-premise solutions, the company was able to substantially reduce infrastructure costs by scaling resources according to demand through the use of AWS's pay-as-you-go model.
- Performance: The distributed nature of AWS EMR enabled parallel data processing, which decreased the time necessary to complete ETL tasks.

In this case study, we look at how Talend, an open-source ETL tool, may be linked with AWS EMR to provide a scalable and cost-effective ETL pipeline for big data.

The project's goal was to handle huge datasets in real time by combining Talend, Spark, and AWS services.

Solution Design

The solution was built to handle data from a variety of sources, including social media networks and enterprise databases. Talend was used to extract data from these sources, format it for analysis, and store it into AWS S3. Spark on AWS EMR was utilized for parallel processing, which allowed the ETL pipeline to manage massive amounts of data.

Results

The Talend-EMR solution effectively processed gigabytes of data in real time, giving the firm significant insights into customer behavior. The solution was both cost-effective (using AWS's pay-as-you-go model) and scalable, allowing the company to process rising volumes of data without sacrificing performance.

Conclusion

This article highlighted how open-source ETL frameworks, when integrated with big data tools and cloud services like AWS, can provide a scalable, cost-effective answer to modern data processing difficulties. Organizations can create strong ETL pipelines that can handle massive datasets in real time by using the strengths of solutions such as Talend, Pentaho, and NiFi. Cloud platforms such as AWS augment these frameworks by providing the infrastructure required for scalability, dependability, and cost-effectiveness.

Future study could look into how machine learning can be integrated into ETL processes to help automate data transformation and increase decision-making skills. As the volume and complexity of data expand, so will the need for scalable, flexible, and cost-effective ETL solutions.

References:

[1] Al-Yadumi, Sohaib, et al. "Review on integrating geospatial big datasets and open research issues." IEEE Access 9 (2021): 10604-10620.

[2] Malik, Piyush. "Governing big data: principles and practices." IBM Journal of Research and Development 57.3/4 (2013): 1-1.

[3] Vassakis, Konstantinos, Emmanuel Petrakis, and Ioannis Kopanakis. "Big data analytics: applications, prospects and challenges." Mobile big data: A roadmap from models to technologies (2018): 3-20.

[4] Phanikanth, K. V., and Sithu D. Sudarsan. "A big data perspective of current ETL techniques." 2016 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2016.

[5] Westmann, Till, et al. "The implementation and performance of compressed databases." ACM Sigmod Record 29.3 (2000): 55-67.

[6] SABTU, ADILAH, et al. "THE CHALLENGES OF EXTRACT, TRANSFORM AND LOAD (ETL) FOR DATA INTEGRATION IN NEAR REALTIME ENVIRONMENT." Journal of Theoretical & Applied Information Technology 95.22 (2017).

[7] Yadav, Harsh. "Scalable ETL pipelines for aggregating and manipulating IoT data for customer analytics and machine learning." International Journal of Creative Research In Computer Technology and Design 6.6 (2024): 1-30.

[8] Rao, T. Ramalingeswara, et al. "The big data system, components, tools, and technologies: a survey." Knowledge and

Information Systems 60 (2019): 1165-1245.
[9] Kwon, Ohbyung, Namyeon Lee, and Bongsik Shin. "Data quality management, data usage experience and acquisition intention of big data analytics." International journal of information management 34.3 (2014): 387-394.
[10] Rao, T. Ramalingeswara, et al. "The big data system, components, tools, and technologies: a survey." Knowledge and Information Systems 60 (2019): 1165-1245.
[11] Mehmood, Erum, and Tayyaba Anees. "Distributed real-time ETL architecture for unstructured big data." Knowledge and Information Systems 64.12 (2022): 3419-3445.
[12] Saecker, Michael, and Volker Markl. "Big data analytics on modern hardware architectures: A technology survey." Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2 (2013): 125-149.
[13] Tang, Shanjiang, et al. "A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications." IEEE Transactions on Knowledge and Data Engineering 34.1 (2020): 71-91.
[14] Berkani, Nabila, Ladjel Bellatreche, and Laurent Guittet. "ETL processes in the era of variety." Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIX: Special Issue on Database-and Expert-Systems Applications (2018): 98-129.
[15] Ullah, Saeed, M. Daud Awan, and M. Sikander Hayat Khiyal. "Big data in cloud computing: A resource management perspective." Scientific programming 2018.1 (2018): 5418679.

