International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Real-Time Data Processing with Streaming ETL

Dhamotharan Seenivasan

Project Lead-Systems, Mphasis, Texas, United States of America
Email: *dhamotharranvs[at]gmail.com*

Abstract: ***Real-time ETL processing using streaming ETL is critical for organizations desiring to utilize up-to-date information and make decisions based on that data. This paper discusses the building blocks, approaches, and advantages of real-time streaming ETL techniques, as well as focuses on structures, tools, and methods by means of which real-time data management could be well-organized and efficient in data processing. The discussion also encompasses real-world examples from different industries globally, with a focus on illustrating the use case and benefits of embracing streaming ETL. Some of the critical issues include data consistency, latency, and fault tolerance, which are discussed with the available solutions and future trends in Real-time data processing. Continuous assessments of data quality and rapid change in data requirements are some ways that make the traditional batch processing methods inadequate to cater for the organizations' need for real-time information, which has led to the integration of streaming ETL systems. These systems involve a constant stream of data processing, whereby information that has not been altered can be converted to tangible results in real-time. Therefore, the goal of this paper shall be to provide a starting point for understanding the architecture, tools, methodologies, and issues of real-time streaming ETL systems.***

Keywords: Real-Time Data Processing, Streaming ETL, Low Latency, Fault Tolerance, Data Ingestion.

1. Introduction

Data real-time processing is no doubt one of the most crucial data processing paradigms of modern organizations. A system that enables data to be processed in real-time is beneficial as it provides timely information, enables decision-making and ready responses to any dynamic environment. This is especially crucial in industries including financial services, pharmaceuticals and retail because real time information can affect result in immediate improvements to processes or client experience.

1.1 Timely Insights

Such insights help businesses quickly respond to trends and events, which is a considerable benefit in terms of competition. For instance, the processing of real-time data is essential in the finance sector to detect and prevent fraud in real time. In this case, through the constant tracking of the transactions and analyzing patterns, the financial institutions can easily detect suspicious activities in as much as within the shortest time possible, hence protecting monetary transactions and maintaining the integrity of the financial transactions. It also helps the customers to safeguard their assets, as well as helping the financial institution become well known for its excellent security measures.

1.2 Improved Decision-Making

Information is essential in decision-making, especially when it is available up-to-date and interferes with a well-functioning organization. It is evident that real-time analytics provides organizations with opportunities to better manage their organization's initiatives, product offerings and customer experiences. For instance, different retail companies can use current information on their operation, such as sales, inventories and customers' feedback, to make real-time decisions on pricing strategies and stock flow and properly align marketing tactics with current market characteristics. These efficiencies can help a business outcompete rivals and serve its consumers more efficiently, thus enhancing agility.

1.3 Streaming ETL

Streaming ETL or Streaming ELT refers to the continuous data ingestion process that involves extracting, transforming, and load processes for real-time data [1]. While the conventional methodology of ETL is designed to deal with data in big lots or, also known as batch ETL, streaming ETL deals with data in real-time as it happens.

1.3.1 Components of Streaming ETL

- **Data Sources:** Streaming ETL can get data from different sources including sensors, logs, databases, APIs, and different IoT devices. These are sources that constantly generate material that must be analyzed in real time.
- **Data Ingestion:** Ingestion tools such as Apache Kafka, Amazon Kinesis, or Azure Event Hub extract the data stream from the sources. They can work as a decoupling layer that both absorbs the incoming information and guarantees its correct transfer to other systems.
- **Stream Processing Engine:** There is a possibility to use stream processing engines from Apache, such as Flink, Spark Streaming, as well as Apache Storm to process the incoming data streams immediately. They enable flexibility in such operations as transformation, updating, selection and consolidation of data within the system.
- **Real-Time Transformation:** This stage is concerned with further processing of the streaming data using several data transformations, including data augmentation data slicing, where unnecessary data can be omitted or data consolidation, whereby several data points can be combined to yield results. These transformations assist in shaping the data to make it ready for analysis and to use it further.
- **Data Loading:** The transformed data is then stored in data storage systems such as SQL/NoSQL databases, data warehouses or in a data lake. These storage systems aid in the provision of a storage environment for the

processed data so that it can be stored for further analysis or even reporting and or visualization.

1.3.2 Advantages of Streaming ETL

- **Real-Time Insights:** Streaming ETL helps organizations derive immediate insights and deliver quick decision making as well as quick responses to dynamic contexts.
- **Reduced Latency:** The analysis of streaming ETL is done in real-time, which means that companies can quickly react to events when they are still happening.
- **Scalability:** As for expansion, streaming ETL systems can scale the solution horizontally, making them able to manage high loads and volumes without compromising performance.
- **Improved Data Freshness:** It is evident that through streaming ETL, organizations engage with the most current data to enhance evaluation and determination.
- **Continuous Processing:** As opposed to batch ETL, which runs on intermittent schedules, streaming ETL works at the click of a button and keeps results as current as the information being fed into the engine.

1.4 Each Component of Streaming ETL

Explanations of Each Component in the streaming ETL process are mentioned in Figure 1.

1.4.1 Data Sources

- **Logs:** Function that records and preserves the log file of the system, applications, and events.
- **Sensors:** Captures data on events occurring in real-time from IoT devices and other sensors and systems.
- **APIs:** A real-time availability of data through different services and applications for their interfaces.
- **Databases:** Data are derived from relational or any other NoSQL DBMS.

1.4. 2 Streaming Data Collectors

- **Apache Kafka:** An extensible, vibrant platform for data consumption and mapping as well as stream processing application creation. The last statement that we have made about Kafka is that it can also be used for large-scale message brokering.
- **Amazon Kinesis:** A subcategory of program outsourcing used for obtaining the means for the organization of the input flows of information and their subsequent further processing.

1.4.3 Stream Processing Engine

- **Apache Flink:** A stateful asynchronous computation model of data streams for unbounded and bounded data stream processing. It is defined as any storage system that is of low latency and high throughput.
- **Apache Spark Streaming:** Apache Spark is a Micro-library that can support careful, high throughput and fault tolerant stream processing of the live data streams.

1.4.4 Real-Time Transformation

- Enrichment: This would include the use of more data, such as joining with the reference data to add more aspects to the data.
- Filtering: Organizing the acquired data may be required by perhaps omitting some details that may not be relevant to the issue at hand or which are not so beneficial.
- Aggregation: Subgrouping large numerical data into relatively denser types like the number of which, how often, total of which for the purpose of analysis.

1.4.5 Data Loading

- **Databases:** Data marts where the transformed data is deposited for consequential query and automated analysis of relational/NoSQL databases.
- **Data Lakes:** Central repositories, which enable you to keep all of your form and non-form data no matter how large.
- **Data Warehouses:** Reporting and analytical systems for read-intensive purposes and data processing, used for the performance of advanced levels of queries.

1.4.6 Use Cases and Applications

- **Real-Time Analytics:** It is possible to analyze new data streams in real-time to arrive at informed decisions.
- **Monitoring and Alerting:** Information can be presented in real time, and events may be set up to initiate an alert for some reason.
- **Fraud Detection:** With the help of such technologies, financial institutions can identify fraudulent transactions within a few moments.
- **Personalized Marketing:** M-commerce services can personalize the experiences of the marketplace in response to user activities in real time.

1.5 Applications of Streaming ETL

1.5.1 Finance

Application of real-time streaming ETL includes real-time fraud detection in finance, high-frequency trading analysis, and risk monitoring. Banks will be able to track transactions in real time and identify the likelihood of the pattern as fraudulent.

1.5.2 Healthcare

In treating patients, streaming ETL is most beneficial in tracking patients' vital signs, as well as any abnormalities in these signs, in real-time. This helps in coming up with correct intervention measures and, as a result, benefits the patient.

1.5.3 Retail

In retail, real-time processing of ETL leads to differentiated and targeted marketing, smarter inventory, and dynamic and smart pricing. Per target consumer metrics, merchandisers can make instantaneous decisions on the best way to reach out to consumers and determine stock supply.

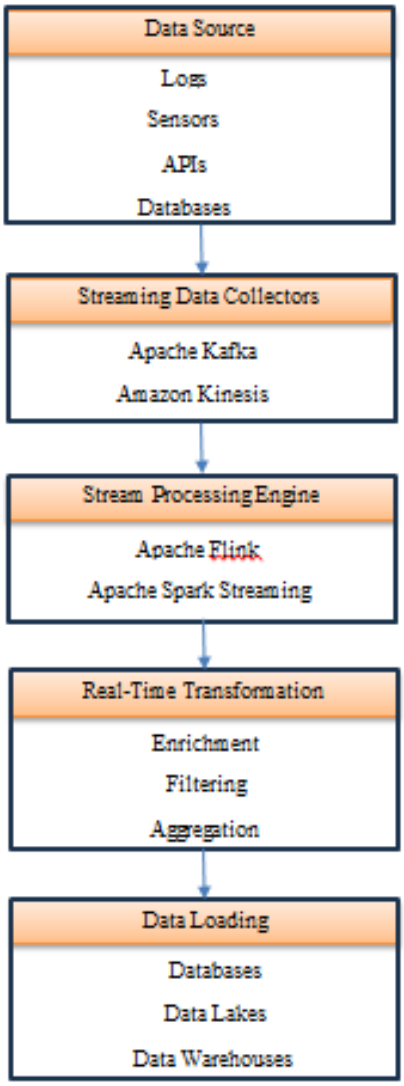

Figure 1: Streaming ETL Component

1.6 Evaluation Metrics

1.6.1 Throughput

Throughput calculates the amount of data to decide on the stream of many records per unit of time.

$$\text{Throughput} = \frac{\text{Total Data Processed}}{\text{Time Taken}}$$

1.6.2 Latency

It is a measure of the time elapsed between the time of data acquisition and the time information arrived at its destination system, impacting the timely delivery of insights.

$$\text{Latency} = \text{Time of Data Arrival} - \text{Time of Data Generation}$$

1.6.3 Data Consistency

Data integrity means that the processed data is good and whole, and there is no loss of data quality from one phase to another.

$$\text{Data Consistency} = \frac{\text{Total Number of Data Instances}}{\text{Number of Consistent Data Instances}} * 100$$

1) Number of Consistent Data Instances: The count of data instances which undergo a correct transformation through the pipeline and retain their integrity.
2) Total Number of Data Instances: The total count of data instances to be processed includes all the data samples of each patient that have been assigned to the target population and will be used for training and testing the model.

1.6.4 Fault Tolerance

$$\text{Fault Tolerance} = \frac{\text{Total Uptime}}{\text{Total Uptime} + \text{Total Downtime}} * 100$$

1) Total Uptime: The sum of the time up until which the system is active and running and has not had a failure.
2) Total Downtime: A measure of the cumulative time in which the system is out of service or not fully functional resulting from failure.

With respect to the Reliability attribute, the feature of the system being able to perform normally even when some components have failed is referred to as Fault tolerance.

1.7 Challenges in Real-Time Data Processing

1.7.1 Data Consistency

One major issue that needs to be addressed is the issue of how to ensure consistency in the case of continuously flowing data. Real-time data processing techniques in organizations require it to handle data from different sources and ensure that the data being processed has not lost its integrity.

1.7.2 Latency

When data needs to be analyzed in real-time, it is important to reduce the time taken before acting on the information. Real-time often means high latency, which can compromise the organization's ability to realize the advantages of such processing.

1.7.3 Fault Tolerance

Ensuring that the system remains available for real-time data processing during such system failures is a critical factor in the overall design of the system. Failure in the hardware and software of a system must not lead to loss of data or make the data inaccessible to GCHQ personnel.

2. Literature Survey

Real-time data processing has arguably emerged as an essential function for organizations that aim to generate real-time insights from their data flows. The traditional ETL (Extract, Transform, Load) processes take time to complete in batch processing mode, while today, data must be mined and turned into insights within real-time, including the concept of streaming ETL, which has become an event-driven process that can analyze data as soon as it is created and address shifting business environments rapidly.

2.1 Evolution of ETL Processes

The shift from batch-centric ETL approaches to streaming ETL is a progression of effective data management approaches. Initially, ETL processes were bi-directional and involved the effectiveness of pulling time-series data at a predefined frequency from systems/ databases, transforming

them and then loading the data warehouses for analytics – making them capable only of historical analysis rather than for real-time decision-making. This led to the emergence of streaming ETL because traditional ETL requires time to pre-process and transform the data to make it usable while streaming ETL allows for near real-time processing of data by businesses and prepares it for use at the appropriate time.

2.2 Batch ETL

ETL is merely a cycle in the conventional operation where information is pulled from a specific data source, changed into another form, and then placed into a data warehouse or any other data repository system [4]. These are mainly created to run one after the other, for example, at certain intervals such as hourly, daily or weekly. This is so because, during each consecutive concrete batch cycle, a large amount of data is collected from different sources, transformed in accordance with definite laws and business logic and then transferred to the target system.

While batch ETL systems work well with big data, the approach inherently introduces more latency between data creation and its availability for analysis. This is so because data is processed only at the end of the batch cycle, meaning high latency. Consequently, while deciding, the analysts must work with a large amount of out-of-date information, which is a significant disadvantage of batch ETL for real-time applications. For instance, the data analysis techniques used for fraud detection, real-time stock price trading or customer engagement become less useful or valuable if analysis is done in batches.

Moreover, the bulk of information that is handled in batch ETL systems can exert pressures on the available resources and rationally in terms of computational intensity and storage capacity during the batch cycles. This leads to inefficiencies and increases operational costs, particularly in organizations with large volumes of data undergoing exponential growth.

Table 1: Comparison of Batch ETL vs. Streaming ETL

Feature	Batch ETL	Streaming ETL
Processing Time	Periodic (hours/days)	Continuous (real-time)
Data Latency	High	Low
Use Cases	Historical Analysis	Real-Time Monitoring
Technologies	Hadoop, Talend	Kafka, Flink, Kinesis
Scalability	Limited by batch size	Highly scalable

2.3 Streaming Technologies and Frameworks

2.3.1 Apache Kafka

Apache Kafka is a distributed stream processing system that is designed to handle these types of data pipelines – those that offer high throughput and low latency [2]. The kind that permits real-time data streaming is industrial and is a feature of many industries. As Kafka is good for handling huge data sets and very fault tolerant, it is one of the most used systems for real time data processing Table 2.

2.3.2 Apache Flink

Apache Flink is a stream processing framework that can support event-driven applications and offers the capability of exactly one application, so data is never duplicated. One of the biggest benefits of using Flink is the fact that its pipelines are designed to support stateful computations and the fact that it is as comfortable when dealing with streaming data as it is when it is working with batches, which makes it an ideal tool for real-time ETL.

2.3.3 Apache Storm

Apache Storm is a distributed real-time computation system which is designed to make real-time parallel processing of large amounts of data and high-velocity data. Facilities like distributed processing, along with support for complex event processing, make it a suitable tool for real-time data analysis for storms.

Table 2: Comparison of Streaming ETL Tools

Tool	Data Ingestion	Data Transformation	Data Loading	Use Cases
Apache Kafka	Yes	Limited	Yes	Log aggregation, Real-time analytics
Apache Flink	No	Yes	Yes	Stream processing, Complex event processing
Apache Spark	No	Yes	Yes	Batch processing, Stream processing
Amazon Kinesis	Yes	Limited	Yes	Data streaming, Real-time applications

2.4 Key Components of Streaming ETL

- Data Ingestion: Collecting current inputs from multiple sources.
- Data Transformation: Performing calculations on the processed data and adding more values to this data in real time.
- Data Loading: Supplying the target systems with the data in their processed state for analysis.

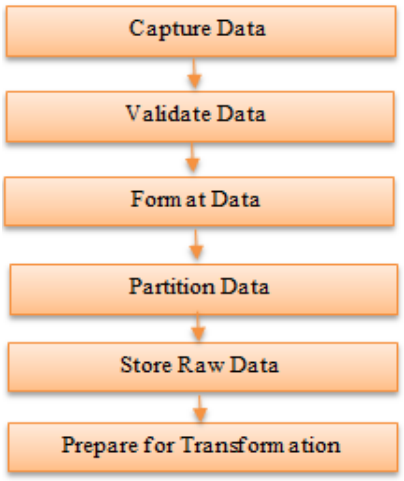

Figure 2: Data Ingestion Process

2.4.1 Data Ingestion Process

The Data Ingestion Process is shown in Figure 2.

a) Capture Data: Gather information from the various platforms, including the Internet of Things, social media platforms, transactions, etc.

b) Validate Data: Accuracy and tidiness of the data being captured by checking for errors, duplications and inconsistencies.
c) Format Data: Transmuting data into forms that are easily manageable and in a format that is easily compatible with the requirements of the computer.
d) Partition Data: Subdivide data into units which can be processed and stored readily In simple terms, data should be segmented.
e) Store Raw Data: Programmatically store the data that is in its raw form before it is processed by the program.
f) Prepare for Transformation: Synchronize and structure the data that is waiting for the transformation to take place.

2.4.2 Data Transformation Process

The Data Ingestion Process is shown in Figure 3.

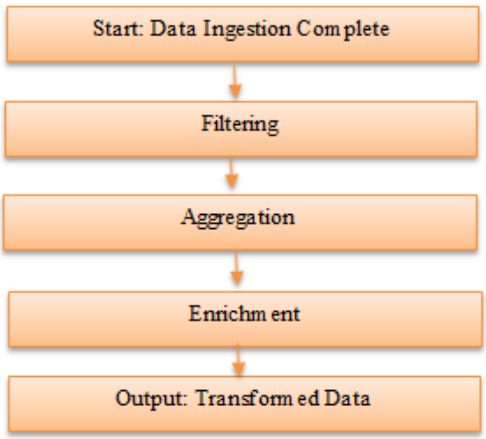

Figure 3: Data Transformation Process

a) Start: The process starts once data has been ingested into a system. Data Analysis Confirmations Given.
b) Filtering: Here, anything that can be considered merging with the crowd and not adding value to the findings is eliminated.
c) Aggregation: This step involves condensing data by summarizing or aggregation so that it can easily be handled while carrying out the analysis. Joining and aggregation are two frequently used operations in relational databases, where joining is a combination of existing data while joining and aggregation operations include calculation of average, sum and count.
d) Enrichment: In the enrichment step, data that has been passed through the filter and aggregation processes outlined above is improved in one form or the other; this could mean, for example, integrating geographical information in the records of sales or demographics on customers into the transaction records.
e) Output: The fourth major step arrives at the transformed data that is ready to be imported into the target system or to be analyzed further.

3. Methodology

3.1 Data Ingestion Techniques in Streaming ETL

The first process of an ETL pipeline, specifically in the streaming platform, is data ingestion, which involves the accumulation of data. Proper collection and management of data is an essential process that must be implemented to guarantee the successful processing of data [3]. The two primary data ingesting methods are as follows;

3.1.1 Log-Based Ingestion

Log-based is a method of ingesting data changes where the change is monitored in the database logs. Figure 4. This makes the technique free from much latency and is highly accurate, especially when dealing with revisions in transactional systems, since it guarantees consistency.

Steps in Log-Based Ingestion:

(a) Data Source: It is the primary data source of material usually derived from the working transactional database.
(b) Database Logs: Files for documenting all the occurrences within the database.
(c) Change Data Capture (CDC) Tool: Analyzing logs of the database and addressing the resulting changes.
(d) ETL Transformation Layer: Takes the transformation of the captured change data.
(e) Data Warehouse/Data Lake: Saves the modified data to data storage units within the system.
(f) Analytics and Reporting: Incorporates the swallowed information into the current and subsequent analysis and reporting.

3.2 API-Based Ingestion

API-based ingestion involves pulling data from Sources using APIs Figure 5. It can help to integrate with web services and IoT devices, thereby facilitating the gathering of data from many sources in real-time and supporting the analytics.

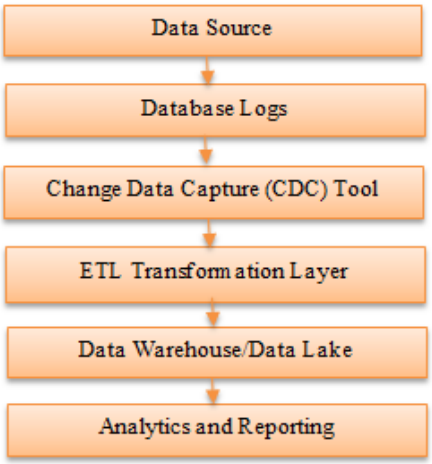

Figure 4: Log-Based Ingestion Flowchart

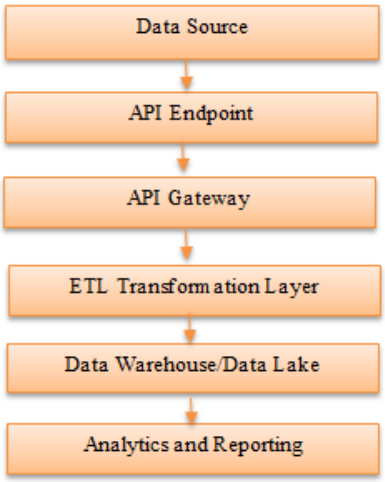

Figure 5: API-Based Ingestion Flows

Steps in API-Based Ingestion;

(a) Data Source: As diverse services available on the web and smart connected things in an environment.
(b) API Endpoint: A feature of a data-processing system that is used to interact with the system and get the desired data.
(c) API Gateway: Manages and directs the API calls and all other business logic in our application.
(d) ETL Transformation Layer: Parenthesize the collected data and transform it.
(e) Data Warehouse/Data Lake: This ensures that the transformed data is stored.
(f) Analytics and Reporting: Runs real-time analytics and generates reports using the ingested data.

Table 3: Comparison of Log-Based and API-Based Ingestion

Feature	Log-Based Ingestion	API-Based Ingestion
Data Source	Transactional systems	Web services, IoT devices
Latency	Minimal latency	Variable depends on API response times
Data Consistency	High	Depends on API reliability
Implementation Complexity	Requires CDC tools, database log access	Requires API integration and management
Use Cases	Tracking transactional changes, maintaining data consistency	Real-time data collection from diverse sources

3.3 Data Transformation Techniques

Online data manipulation means the data undergoes transformation once it is being fed into the system to make the data suitable for analysis. Below are the main techniques used for real-time data transformation:

3.3.1 Stream Processing

Apache Kafka Streams and Apache Flink are employed for streaming processors in a way. In Figure 6, the actual real-time transformations among the fields include filtering, aggregating, and enriching the values. These are structures that are intended for consuming steady data flows to achieve real-time analytic capabilities.

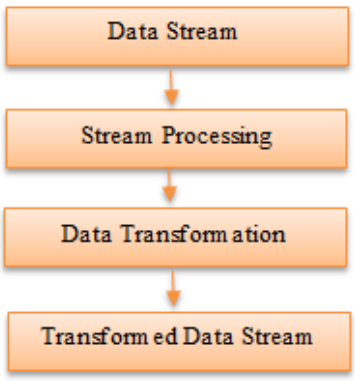

Figure 6: Stream Processing process

a) Data Stream: Distribution or incoming data stream received from different origins.
b) Stream Processing Framework: There are similar applications, Apache Kafka Streams, Apache Flink and many more.
c) Data Transformation: Analyses such as filtering, aggregation or enrichment in real-time.
d) Transformed Data Stream: Data flow output with data ready for analysis after the data has been transformed.

3.3.2 Complex Event Processing (CEP)

CEP is also defined as the method for identifying patterns and relationships in the flow of the data to avoid or to prepare for event occurrences in Figure 7. This technique is very applicable to various applications such as fraud detection and other types of anomaly detection, where data has to be analyzed in real-time.

(a) Data Stream: Endless flow of information from all departments and other sources.
(b) CEP Engine: Contains a brain that is capable of processing complicated events and Stream 1 patterns.
(c) Pattern Recognition: Searching for the relationships of the data flow.
(d) Event Correlation: Event coincidence with different data streams.
(e) Action Triggering: Applications that can invoke certain actions or cause-specific alerts as per the determined patterns and correlations.

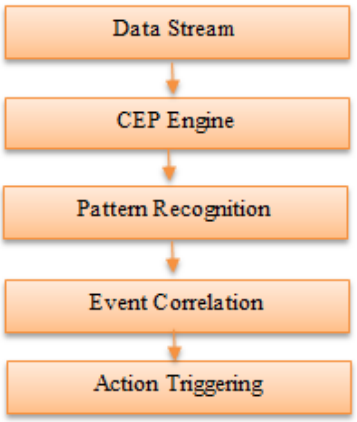

Figure 7: Flowchart of CEP

3.4 Data Loading Techniques

The last stage in a streaming ETL workflow is the load process, Table 4, which transfers the data that has been transformed in the preceding phase into the target systems. Below are the main techniques used for data loading

3.4.1 Data Warehouses

Service-based architectures from vendors like Amazon and Google that use real-time data warehouses like Amazon Redshift and Google Big Query provide low latency for queries. Such systems present an opportunity for scalability and performance relevant to real-time data analysis, thus allowing organizations to derive insights from hefty information mashups.

3.4.2 Data Lakes

A data lake can be described as an infrastructure featuring high scalability and designed for the storage of original and processed data for use in different forms of analysis and machine learning. They allow organizations to capture and process large amounts of data without needing to be strictly pre-arranged; they are useful for data capture and processing in real-time.

Table 4: Comparison Table of Data Warehouses and Data Lakes

Feature	Data Warehouses	Data Lakes
Storage Structure	Structured data	Structured, semi-structured, unstructured data
Query Performance	Low-latency queries	Variable performance based on data lake architecture
Schema Flexibility	Limited schema flexibility	Schema-on-read approach, flexible schema definition
Analytics Support	SQL-based analytics, real-time querying	Support for various analytics and machine learning workloads
Use Cases	Real-time analytics, business intelligence	Ad-hoc analysis, exploratory data science

4. Results and Discussion

4.1 Results of Real-Time Data Processing with Streaming ETL

4.1.1 Improved Data Freshness and Latency Reduction

Real-time ETL has less latency than batch ETL systems because of Streaming organization. Here is a table comparing batch ETL and streaming ETL data latency.

System Type	Average latency (minutes)
Batch ETL	1440 (1 day)
Streaming ETL	5

4.1.2 Enhanced Fraud Detection

With the use of streaming ETL in identifying fraud, financial institutions have noted an advancement in the time that it takes to detect the same.

Metric	Batch ETL	Streaming ETL
Detection Time (avg)	1-2 hours	5-10 minutes
Fraudulent Transactions Detected (%)	75	95

4.1.3 Optimized Inventory Management in Retail

Correspondingly, the application of streaming ETL in retail has improved the inventory control approach that has negatively impacted stockout and stock-up scenarios.

Inventory Metric	Before Streaming ETL	After Streaming ETL
Stockouts per Month	15	5
Overstock Instances	20	8

4.1.4 Real-Time Health Monitoring

This can be seen clearly in the case of high-level patient monitoring in which Streaming ETL has contributed towards the early identification of anomalies.

Metric	Traditional Monitoring	Streaming ETL
Anomaly Detection Time	30 minutes	2 minutes
Response Time Improvement (%)	50	90

4.2 Discussion on Streaming ETL Approaches and Challenges

4.2.1 Architectural Components and Flexibility

A streaming ETL system comprises several elements like data feeds, ingestion tools (Apache Kafka, Amazon Kinesis Data Streams), stream processing technologies (Apache Flink, Apache Spark Streaming), and data storage solutions (data lakes and warehouses). These systems entail modularity, and this is a strength as they can be easily scaled up or down as needed.

4.2.2 Scalability and Fault Tolerance

Metric	Streaming ETL
Scalability (Data Volume per Second)	10,000+
Fault Tolerance	High

4.2.3 Challenges in Data Consistency and Integration

Duplicated data make it difficult to maintain orthogonality among various data sources and during data transformation processes. They maintain that organizations require highly complex algorithms and validation processes.

Challenge	Description
Data Consistency	Ensuring consistent data across sources
Integration Complexity	Combining data from diverse sources

4.2.4 Latency Considerations

They consist of ingestion tools, stream processing engines, and storage, and all of these must be fine-tuned to provide low latency.

Latency Factor	Impact on Performance
Ingestion Tools	High
Processing Engines	High
Storage Solutions	Medium

4.3 Future Trends and Developments

4.3.1 Integration with Machine Learning

The incorporation of real-time ML algorithms in ETL processes also improves predictive analytical computations. For instance, first-party retailers can make recommendations in real time to first-party customers.

4.3.2 Edge Computing

Edge computing can similarly help decrease latency even additionally by processing information near the source. This is advantageous, especially in fields such as self-driving cars and the use of drones in industries.

4.3.2 Advancements in Streaming Frameworks

Current issues present in streaming frameworks will be solved in future developments of the frameworks to make the infrastructure more effective for real-time data processing.

5. Conclusion

Streaming ETL as a new model of real-time data processing is paving the way to revolutionizing the ways data is collected and analyzed within an organization. Streaming ETL systems offer a way of constantly feeding data into the proper structures to be analyzed so that organizations can gain valuable insights required to make proper decisions and or enhance customer experience. For example, in e-commerce, real-time data can be used immediately to promote highly relevant products to consumers; in the case of financial services, fraudulent activities can be pinpointed and prevented as they are initiated on the internet. The ability to process data and analyze them in real time is a particularly useful asset in a world that increasingly boils down to a singular question – timing. The fact that streaming ETL is dynamic makes it beneficial to the business since it assists in corporate agility and competitiveness in rapidly changing markets.

Several demerits are in equal proportion to the streaming of ETL, as it has been noted above. Although there can be many sources of data originating from various systems, it is difficult to update such sources with fresh, constant data during specific intervals. To do this is necessary to use complex algorithms and robust IT support. By minimizing the latency, it is always necessary to stay as close as possible in real-time, and here, one needs powerful processing solutions. In addition, it is crucial to design the system so that it does not fail at each checkpoint in a manner that causes loss of data while also making the system robust. Thankfully, current advancements in the frameworks and platforms used in streaming, such as Apache Kafka and Apache Flink, have been offering solutions to these issues, enhancing the efficiency of systems dealing with real-time data. Subsequent studies should aim at developing and fine-tuning such systems, finding solutions to the issue of data confidentiality, as well as exploring the possibilities of using these in different fields of industry and commerce. Further on, with the dissemination of technology, the real-time streaming ETL will remain be valuable approach towards the analysis of data in the future and affect new generations as one of the major components in the analysis and solution of issues of the data-oriented society.

References

[1] What Is Streaming ETL?, Hazelcast. https://hazelcast.com/glossary/streaming-etl/
[2] Dhamotharan Seenivasan, "ETL (Extract, Transform, Load) Best Practices," International Journal of Computer Trends and Technology, vol. 71, no. 1, pp. 40-44, 2023. Crossref, https://doi.org/10.14445/22312803/IJCTT-V71I1P106
[3] Key Features of Leading Stream Processing Frameworks, Risingwave. https://risingwave.com/blog/inside-look-exploring-the-best-stream-processing-frameworks-of-2024/
[4] What is Data Ingestion?, Qlik. https://www.qlik.com/us/data-ingestion
[5] Dhamotharan Seenivasan, "Exploring Popular ETL Testing Techniques," International Journal of Computer Trends and Technology, vol. 71, no. 2, pp. 32-39, 2023. Crossref, https://doi.org/10.14445/22312803/IJCTT-V71I2P106
[6] ETL Batch Processing: A Comprehensive Guide, Astera. https://www.astera.com/type/blog/etl-batch-processing/#:~:text=ETL%20batch%20processing%20involves%20handling,processes%20it%20as%20a%20batc
[7] Dhamotharan Seenivasan, "Improving the Performance of the ETL Jobs," International Journal of Computer Trends and Technology, vol. 71, no. 3, pp. 27-33, 2023. Crossref, https://doi.org/10.14445/22312803/IJCTT-V71I3P105
[8] https://www.confluent.io/learn/etl-elt-streaming-data-compared/
[9] Dhamotharan Seenivasan, Muthukumaran Vaithianathan, 2023. "Real-Time Adaptation: Change Data Capture in Modern Computer Architecture" ESP International Journal of Advancements in Computational Technology (ESP-IJACT) Volume 1, Issue 2: 49-61.
[10] https://www.deltastream.io/what-is-streaming-etl-and-how-does-it-differ-from-batch-etl/
[11] https://www.jumpmind.com/blog/blog/data-trends/what-is-streaming-etl-2022/
[12] https://www.decodable.co/blog/the-top-5-streaming-etl-patterns