International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 2, Issue 3, PP.74-81, 2021
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I3P109

Review of Streaming ETL Pipelines for Data Warehousing: Tools, Techniques, and Best Practices

Vaibhav Maniar[1], Rami Reddy Kothamaram[2], Dinesh Rajendran[3], Venkata Deepak Namburi[4], Aniruddha Arjun Singh Singh[5], Vetrivelan Tamilmani[6]
[1]Oklahoma City University, MBA / Product Management.
[2]California University of Management and Science, MS In Computer Information Systems.
[3]Coimbatore Institute of Technology, MSC. Software Engineering.
[4]University of Central Missouri, Department of Computer Science.
[5]ADP, Sr. Implementation Project Manager.
[6]Principal Consultant (SAP), Infosys Ltd.

Abstract: The fast generation of data-based applications has intensified the burden on the necessity to execute data integration into the warehousing systems in real-time and efficiently. Extract, Transform, Load (ETL) based streaming pipelines have emerged as a key solution choice, and continuous data ingestion, transformation and delivery of data, and timely analytics and decision making can be facilitated. The authors in this review seek to examine the principles and underlying techniques, tools and best practices which enable streaming enabled architectures where special focus is directed towards how they enable scalability, elasticity and fault tolerance within dy3namic data ecosystems. Stream processing models, data coordination schemes and data consistency and quality assurance mechanism of near-real-time processes have been found to have the most significant influence. Implementation in other fields of streaming ETL is also discussed in the paper whereby it has proved to save processing latency, enhance operational efficiency, and, in addition, enhance the reliability of the analytical results. The survey is an elaborate review of the transformation of the current practices of integrating real-time data through the incorporation of the new developments and applications. The results have advice on developing adaptive, intelligent and sustainable data warehousing systems that have the potential to cope with the growing demand of the contemporary businesses and assist the next-generation analytics programs.

Keywords: Streaming ETL, Data Warehousing, Real-Time Analytics, Data Integration, Big Data, Scalability, Fault Tolerance.

1. Introduction

The modern world of data is more dependent on timely, accurate, and actionable information to support strategic decision making than at any other time ever [1]. Although traditional data warehousing can be useful in aggregating past data to analyze them, it lacks the ability to respond to swiftly changing world with its rapid-moving, volatile nature. As the volume of data has exploded with social media, IoT, financial transactions and online shopping, organizations have moved out of a batch and extract-transform-load (ETL) world into the streaming ETL pipeline era, which supports the flow of data and insightful actions on moving data rather than a snapshot data map. Streaming ETL is an essential connector between the real-time data generated within the organizations and analytics with regard to the context of data warehousing systems. At implementation, streaming ETL ingests, transforms, and loads data in real-time to have low-latency decision support available, and timeliness of operations, in a more timely fashion [2]. Streaming ETL pipelines enhance data freshness, and enable businesses to prioritize valuable intelligence work, such as fraud detection, customizing customers, and preventive maintenance, using the most recent data. Using and being able to draw intelligence out of real-time data is a pre-requisite to success when industries start operating fast within fast-moving digital ecologies.

The shift between the old ETL and streaming ETL might be attributed to the major developments in the fields of distributed computing models and real-time data processing tools. Along with the introduction of Apache Kafka, Apache Flink, and Apache Spark, there were also high-throughput data ingestion and transformation paradigms and real-time stream processing at scale that can also be fault tolerant [3]. Along with the invention of platforms to process real-time stream, the best practice of data quality and fault tolerance was developed to tackle the issue of updating data in real-time environment. According to these changes, the existing data warehouses need to redefine the traditional ETL architecture. Streaming ETL pipelines are also future but there are several challenges that come along with their implementation. Of concern is the quality and consistency of data on motion and it is especially important when one is handling heterogeneous and accelerated sources of data [4]. It should also be fault tolerant, scalable, elastic and secured to ensure that failures and inaccuracies are avoided. Finally, a gap in research on the establishment of standard frameworks to support various enterprise architects exists, and that is the reason as to why the systematical review of the tools, techniques and best practices at the disposal of researchers and practitioners is necessary.

Through an in depth and systematic study of the streaming ETL pipelines as a support of data warehousing by exploring the tools, techniques and best practice identified in the research literature, this review seeks to add to a better understanding of how

streaming ETL can help in changing the nature of data warehousing to become a responsive and real time decision support environment. [5]. This review has touched on the change of traditional ETL to real-time data pipeline, data quality and fault-tolerant structures, and the architecture to support the design of scalable and elastic systems. This review brings together the current research contributions, explains the significant and topical challenges and proposes the ways of further investigation.

1.1. Structure of the Paper

The paper is structured as follows: Section II covers streaming ETL fundamentals and real-time data characteristics; Section III examines best practices for scalability, fault tolerance, and data quality; Section IV reviews key streaming ETL tools and techniques; Section V summarizes the literature's findings, challenges, and future directions; and Section VI concludes with insights and recommendations.

2. Streaming ETL in Data Warehousing

A Streaming ETL engine, in the context of Streaming DW, is a system of interrelated nodes that, when applied to data obtained from streams, performs actions analogous to those of the traditional ETL phase. The Streaming ETL engine's structural elements are:

- **Remote Buffer Framework (RBF):** Acts as a buffering layer for incoming data streams, decoupling producers and consumers while ensuring smooth flow and preventing overload.
- **Remote Integrator Framework (RIF):** Integrates and synchronizes heterogeneous data streams, performing normalization and fusion to create a unified real-time data flow.
- **ELT-RT:** A real-time version of the ELT process, where data is extracted and loaded immediately, with transformations applied on the fly for low-latency processing.
- **Fault Tolerant Integrator (FTI):** Provides resilience by handling failures through checkpointing, replication, and recovery mechanisms to guarantee reliable and consistent stream processing.

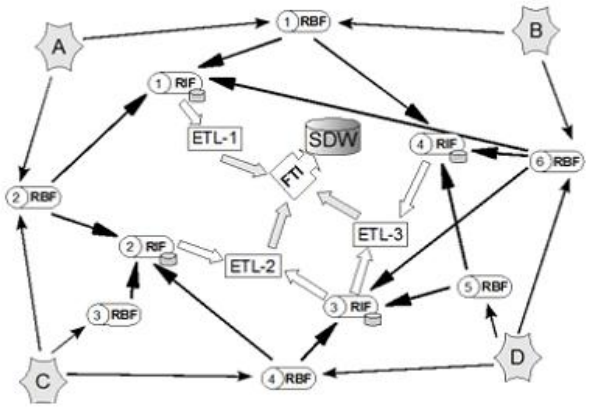

Figure 1: An example of Streaming ETL architecture

The Streaming ETL engine is depicted in Figure 1 as a network, with objects labelled with capital letters A through D standing for data streams. With the help of at least one RBF component, all of the sources can instantly receive data from the tethered Streaming DS. Next, RIF modules handle tuples by retrieving data from linked RBFs, storing it, and subsequently enabling ETL-RT modules to download it. One major benefit of RIF components is their ability to hold tuples collected from several RBFs along with the identities of the input streams. Thus, RIF modules can be shifted between source RBFs in the event of an engine failure. Last but not least, the Streaming ETL engine's Fault Tolerant Integrator (FTI) checks for mistakes, compares duplicate data streams, and combines them into one. This allows the processed data to be fed into the Streaming DW [MAL]. Therefore, the RIF module in an ETL-RT implementation responsible for inserting data.

2.1. Evolution from Batch ETL to Streaming ETL

Conventional ETL (Extract, Transform, Load) pipelines were designed for batch-oriented data processing, in which data would be collected, transformed, and loaded in scheduled intervals for generating business intelligence reports [6]. The conventional method was suitable for static environments. Its problematic conditions included latency, rigidness, and the inability to act on high-velocity, continuously incoming data. As applications demanded faster insights, the concept of streaming ETL, which evolved from batch-oriented ETL, embraced both high-velocity data loads and real-time insights. Streaming ETL supports processing data in motion and generating actionable insights at ultra-low latencies. Streaming data processing is suitable for use cases such as fraud detection, IoT sensor monitoring, and clickstream data analysis. Streaming ETL helped pave the way for the concepts of real-time data warehousing and analytics, utilizing distributed computing frameworks, messaging systems, and architectural models such as Lambda and Kappa, which were eventually realized and adopted.

2.2. Characteristics of Real-Time Data in Warehousing

Real-time data in warehousing is defined by its ability to deliver insights instantly as data is generated, ensuring timely decision-making and continuous analytics [7]. Unlike batch data, it is dynamic, fast-moving, and requires specialized handling to maintain accuracy and performance.

- **High Velocity** – Data arrives continuously from multiple sources such as sensors, applications, and user interactions.
- **Low Latency Requirement** – Processing and loading must occur with minimal delay to support instant insights.
- **Volume and Variety** – Structured, semi-structured, and unstructured datasets of varying sizes and types stream data warehouses.
- **Continuity** – Data streams are ongoing and do not follow fixed intervals like batch processes.
- **Data Volatility** – Information changes rapidly, requiring immediate transformation and integration.
- **Scalability Need** – Systems must dynamically scale to accommodate unpredictable data surges.
- **Accuracy and Consistency** – Despite speed, data integrity and correctness must be preserved in motion.
- **Fault Tolerance** – Pipelines must recover from failures without losing or duplicating records.

2.3. Real-Time ETL Adoption

Several key factors have driven organizations to adopt Real-time ETL processes [8]:

- **Demand for Immediate Insights:** Modern data is crucial for many businesses to have an edge in the market, provide better service to customers, and run efficient operations [9]. For instance, fraud detection in financial services, inventory tracking in retail, and patient monitoring in healthcare all require data to be analyzed as it is generated. Traditional batch ETL introduces latency, limiting the utility of the data for real-time applications.
- **Proliferation of Real-Time Data Sources:** The growth of IoT devices, mobile applications, and online transactional systems has increased the volume of real-time data streams. These data sources continuously generate data, necessitating a system that can ingest, transform, and store information without delay.
- **Advances in Technology:** Modern advancements in computing power, distributed processing, and cloud infrastructure have made real-time ETL feasible. Technologies such as Apache Kafka, Spark Streaming, and cloud-based data pipelines have provided the foundation for real-time data integration, allowing data to be processed in motion rather than waiting for periodic batch jobs.
- **Improvement in Analytics and Business Intelligence (BI):** As BI tools evolve, they increasingly support Real-time analytics, which relies on real-time data availability. Businesses are shifting from retrospective, batch-based reporting to proactive, real-time insights, leveraging data for immediate decision-making and action.

3. Streaming ETL Techniques and Tools

Streaming ETL methods enable continuous data ingestion, transformation, and loading in real-time, with low latency and high availability for analysis. An unlimited data flow from many sources, including IoT devices, logs, and social media streams, is processed by it, setting it apart from typical batch ETL methods [10]. Some of the key techniques available in streaming ETL are micro-batch processing, windowed aggregation, and event-driven processing (push-based), all of which enable continuous data processing and make data available to a data warehouse sooner. Tools that can be used in streaming ETL include Apache Kafka, which is great at ingesting high-throughput data, and Apache Spark Streaming and Apache Flink can handle streaming data in a distributed way that is also fault-tolerant and can scale, like Apache NiFi, which can orchestrate data using flow management that is also capable of integration with lots of data sources [11]. These various tools allow streaming ETL and streaming analytics to break down the usual impediments to analytics and allow organizations to make more data-driven decisions in less time.

3.1. Data Ingestion Techniques

Amazon Kinesis, Google Pub/Sub, Apache Kafka, NATS Streaming, Microsoft Event Hubs, and Google [12] are among of the most used solutions for commit logs. Streaming data frameworks like this find use in log metrics, web services, workplace apps, the IoT, and driverless vehicles [13]. A strong, well-structured data streaming infrastructure is the only way to handle the massive amounts of data processing required by the recently popularized artificial intelligence. Here takes a quick look at NATS, Apache Kafka, and RabbitMQ MQ, and then compare and contrast all three of these frameworks.

3.1.1. NATS Streaming

Cloud-based messaging system NATS, developed and maintained by Synedria Group, is lightweight and open-source. It is implemented in the Go programming language [14]. This approach is compatible with messaging queues, request-reply, and publisher-subscriber interactions. Members of the NATS community can communicate with one another through the pub/sub system. Only receive messages if actively listening to these issues. One way the NATS server, or gnats, helps with scalability is by disabling subscriptions in the event that a connection timeout happens.

3.1.2. Apache Kafka

The Scala implementation of Kafka makes it a durable, scalable, and fault-tolerant publish-subscribe messaging system. It is used by several top firms, including LinkedIn, Yahoo!, Twitter, and more [15]. Although real-time analytics data streaming is

Kafka's primary use case, the platform has several other potential uses as well, including monitoring, message replay, log aggregation, error recovery, and website activity tracking. With its user-friendly interface, fast throughput, and reliable replication feature, Kafka is an excellent choice.

3.1.3. RabbitMQ

An application that mediates between different apps is known as a message broker. Among these message broker systems, RabbitMQ is one. The Advanced Message Queuing Protocol (AMQP) is a part of it, and it's open-sourced. Asynchronous message-based communication between apps is made possible with its help. There is no requirement for the sender and recipient systems to be operational at the same time because the messages are loosely connected.

3.2. Real-Time Data Warehousing and ETL Techniques

Decisions based on operational data that may be made by businesses in near real-time have become increasingly important. Conventional ETL systems that priorities batch updates cause substantial delays in updating the data warehouse [16]. Raising doubts about the effectiveness of batch ETL designs and near-real-time updates on operational data highlights the need for real-time research to support rapid business choices. In spite of difficulties with view synchronization and resource allocation brought on by online data warehouse updating, existing ETL procedures must change from periodic refreshes to continuous updates. "The data warehouse needs continuous data integration capabilities to handle the most recent business data and meet real-time requirements." In indiscriminately updating data from various sources, view synchronization concerns emerge. Conflicts between queries for long-term analysis and changes made at the same time might lead to resource constraints. There is no way to access the data warehouse while using traditional ETL solutions because they load data periodically during downtime [17]. This partition makes some parts of data warehouse installation easier, but it also prevents the warehouse from being updated constantly. Continuous updates without downtime are beyond the capabilities of traditional solutions.

3.3. Tools for Implementing Data Quality Checks

Data quality checks in ETL pipelines can be facilitated by a variety of tools, including both commercial and open-source options [18]. These tools provide the capabilities of data profiling, cleansing, monitoring and validation properties that allow organizations to perform effective data quality checks.

- **Informatica Data Quality:** A complete data quality system that provides profiling, cleaning and tracking of data quality. It supports the combination of data with set of ETL tools provided by Informatica, that provides a convenient approach to data quality maintenance.
- **Talend:** An open-source data integration platform and contains data quality features. Talend aids in profiling of data, cleaning and approving data, and enabling companies to offer verifications of the quality of data into their ETL procedures.

IBM Infosphere Quality Stage: Quality Stage belongs to the Infosphere family provided by IBM and it provides the possibility of data profiling, cleansing and matching. It also has become part of IBM ETL tools to help in the general control of the data quality.

Bioenergy refers to electricity and gas that is generated from organic matter, known as biomass. This can be anything from plant and timber to agriculture and food waste and even sewage. Bioenergy includes the production of fuel from organic matter as well. Energy from biomass can be used for electricity, heating, and transportation, and can be replenished anywhere. Around seventy-five percent of the world's renewable energy is composed of biomass energy due to its potential and wide use [7]. Also, it is carbon-neutral, meaning that it adds no net carbon dioxide to the atmosphere. In addition, it reduces the level of trash in the ground by as much as 90 percent by burning solid waste. Biomass fuels, on the other hand, are not completely clean and can also cause deforestation. They are also less efficient than fossil fuels. But proper management and planning of its disadvantages will improve its potential. Bioenergy refers to electricity and gas that is generated from organic matter, known as biomass. This can be anything from plant and timber to agriculture and food waste and even sewage. Bioenergy includes the production of fuel from organic matter as well. Energy from biomass can be used for electricity, heating, and transportation, and can be replenished anywhere. Around seventy-five percent of the world's renewable energy is composed of biomass energy due to its potential and wide use [7]. Also, it is carbon-neutral, meaning that it adds no net carbon dioxide to the atmosphere. In addition, it reduces the level of trash in the ground by as much as 90 percent by burning solid waste. Biomass fuels, on the other hand, are not completely clean and can also cause deforestation. They are also less efficient than fossil fuels. But proper management and planning of its disadvantages will improve its potential.

4. Best Practices in Streaming ETL Pipelines

An ETL system refers to a system that is used by an organization to depict flow of data and a transaction process that are used to pull data out of a wide array of data sources and communication protocols [19]. This involves the need of having

knowledge to design these processes in a manner that would be efficient in aspects of acquiring data, transforming data and loading of the data into data storage environments. The ETL process is significant aspect of the data management practices like integration and migration of the data. It involves gaining access to the data in different locations of origin, processing the data through cleaning, combining and staging and presenting the processed data to the storage systems.

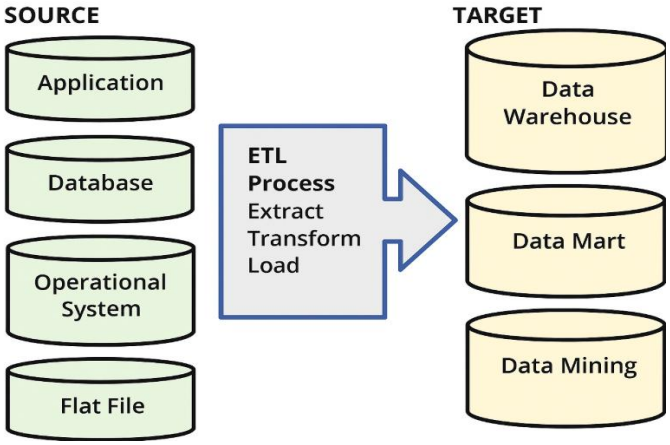

Figure 2: ETL System Flow in a Data Warehouse Environment

Figure 2 illustrates the ETL process. Data warehousing and data marts are examples of destination systems that undergo data extraction, transformation, and loading. Data can originate from a variety of sources, including applications, databases, operational systems, and flat files. This ensures integrated storage and helps with efficient decision-making.

4.1. Scalability and Elasticity in Real-Time Workloads

When working with real-time ETL workloads, scalability and elasticity are fundamental properties. This is because data streams can be quite ephemeral and dynamic. Scalability refers to a system's ability to effectively scale up to meet the workload, increasing the amount of processing power, storage, and/or network capacity as needed, thereby ensuring that performance does not degrade as data volume and velocity grow [20]. Elasticity is the landscape feature that allows resource levels to dynamically and automatically scale up and down to suit real-time demand. This characteristic prevents over-provisioning and helps relieve operational costs by allowing systems to scale resources down during times of low traffic. Together, these properties are vital features of streaming pipelines, ensuring low latency, high throughput, and reliability. In the context of modern data warehousing, the ability to be scalable and elastic provides the underlying infrastructure to support the ingestion, transformation, and loading of continuous data, thereby driving the demands of an analytical system in real-time. Again, both specifications support analytical systems that serve their clients, even with varying workloads and sudden traffic spikes.

4.2. Fault Tolerance and Recovery Mechanisms

There are three distinct phases to fault tolerance mechanisms: Systems, Applications, and Hardware Failure [21].

- **Hardware Fault Tolerance:** The supply of additional hardware, including central processing units, memory, hard drives, power supply units, etc., is part of this. Tolerating hardware failures only allows for the provision of a minimal hardware backup system; it is unable to prevent or detect errors, unintended program interference, program errors, etc. Computer systems that automatically fix hardware component failures are constructed according to the principles of hardware fault tolerance.
- **Software Fault Tolerance:** This is specialized software that can withstand mistakes that could be caused by programming or software flaws [22]. Like hardware fault tolerance, software fault tolerance makes use of static and dynamic redundancy techniques. An N-version programming strategy utilizes the same concept as a single program that uses static redundancy, whereby programs that fulfill an identical purpose are being selected at certain checkpoints.
- **System Fault Tolerance:** This system is more than merely saving checkpoints; it automatically identifies application errors, blocks as well as memory and programs. There is a built-in error correction mechanism in the system, and hence it is capable of correcting errors once they occur.

4.3. Handling Data Quality and Consistency in Motion

Ensuring data quality and consistency is a massive issue in real-time ETL pipelines. All of them are concerned with missing or delayed records, and some uncertainty with the incoming information of other kinds, different rules, schemas, and inconsistencies. In batch systems, the processes required to guarantee the data validation maybe done later after the ingestion or delayed to later when the data meets specific validation requirements [23]. In streaming types of environments, that option is not an option because the data is flowing and has to be processed and delivered within some low-latency threshold. In real-time ETL pipelines, data quality relates to filtering out duplicates, managing delayed or missing records, handling schema changes, and validating incoming values. Data consistency entails whether the order of updates identified from "events" is verified in-stream

and whether downstream systems are always presenting the actual state of the data [24]. Techniques such as stream deduplication, event time ordering, watermarking and schema evolution are commonly applied approaches, allowing businesses to establish and maintain credible and authoritative insights, reduce the occurrence of outliers, and provide trustworthy real-time analytics for operational and tactical decision-making.

5. Literature of Review

The reviewed studies collectively emphasize advancements in streaming ETL pipelines, addressing architectural evolution, ingestion frameworks, real-time analytics, and query-driven maintenance, while highlighting persistent challenges of scalability, latency, and integration. They also propose future directions for robust, adaptive, and intelligent data warehousing solutions. Alam and Kamal (2019) a discussion of data warehousing, beginning with the conventional model and progressing to real-time data warehousing, as well as the societal effects of the latter. Topics related to data warehouse architecture are also included in this survey. The document explains how the extract-transform-load method has been modified to accommodate real-time data warehousing. There is no real-time, near-real-time, or today-data in the data warehouse that it uses to draw integration data. The typical timeline for data loading into a standard data warehouse is either weekly or nightly, and it can come from a single or numerous operation sources [25].

Katari (2019) identified the core architectural elements essential for creating ETL pipelines that can handle the dynamic and fast-paced nature of financial data. Real-time financial analytics demands an architecture that can ingest large volumes of data at high speeds while ensuring data accuracy and integrity. Key architectural considerations include selecting data storage solutions, integrating scalable and flexible data processing frameworks, and implementing low-latency data transformation techniques. Additionally, the architecture must support fault tolerance and data security, given the sensitive nature of financial information. Challenges in building such ETL pipelines are multifaceted [26]. Isah and Zulkernine (2018) outline the existing landscape of data stream ingestion systems, presents a scalable and fault-tolerant framework for data stream ingestion and integration, and demonstrates its practical application in a real-world data stream processing case study. The case study utilizes Apache NiFi and Kafka to process global quick news articles. In addition to outlining current best practices, the report pinpoints areas where more research is needed to establish infrastructure for processing data streams on a wide scale [27].

Chandra and Gupta (2017) data warehousing (DW) has captured the interest of both researchers and businesses. There have been a lot of papers published in the last few years discussing different problems and difficulties in data warehousing. The purpose of this article is to offer a thorough overview of the current research trends in data warehousing by means of a survey. Because information can be retrieved from data through processing, and knowledge can be derived from data and information through analysis, both data and knowledge play crucial roles in a wide range of human endeavors [28]. Minhaj (2016) data warehouses have long been a desired feature for business intelligence customers, but the traditional ETL procedure does not allow for real-time insights because the warehouse updates its data offline and in batches. A potential solution to this problem is the rise of near-real time ETL techniques. This paper aims to investigate the main strategies developed for near-real-time ETL after first looking at the practical issues with the traditional ETL method that are leading to data latency in data warehouses [29].

Qu et al. (2015) snapshot is where the deltas are located on the source side of ETL flows. The initial stage in responding to a query is to update relevant tables with the exact source deltas obtained at the time of the inquiry (also known as query-driven policy). Updating snapshots is handled via an incremental recompilation pipeline that is cleared out by a series of consecutive deltas from incoming queries. To produce a serializable schedule of concurrent maintenance operations and OLAP queries, a workload scheduler is utilized. Utilizing workloads that are high on reads and updates has allowed us to analyses performance [30].

Table I summarizes key studies on streaming ETL pipelines, outlining research focus, methodologies, major findings, challenges, and future directions, providing a comprehensive overview of advancements in data warehousing practices.

Table 1: Summary of a Study on Streaming ETL Pipelines for Data Warehousing

Author	Study On	Approach	Key Findings	Challenges	Future Directions
Alam & Kamal (2019)	Real-time data warehouse evolution	Survey from traditional DW to real-time DW architectures	Real-time DW enhances decision-making and societal impact	Integrating real-time data streams with legacy systems	Improve ETL processes for seamless real-time integration
Katari (2019)	ETL pipelines for financial analytics	Architectural survey of real-time financial ETL	High-speed ingestion ensures accuracy & integrity in financial data	Low latency with fault tolerance & security	Enhance secure, scalable, low-latency ETL frameworks
Isah & Zulkernine	Ingestion systems for	Efficient and reliable ingestion	Demonstrated real-world high-velocity ingestion of	Handling heterogeneous	Standardizing reusable ingestion

(2018)	data streams	platform utilizing Apache NiFi and Kafka	unstructured/structured data	input sources at scale	frameworks
Chandra and Gupta (2017)	Research trends in data warehousing	Comprehensive survey of DW issues	DW research has broad applications in decision-making and knowledge extraction	Scalability, adaptability, and evolving data needs	Holistic frameworks integrating AI/ML for DW
Minhaj et al. (2016)	Near real-time ETL	Exploratory study of batch vs. near real-time ETL	Near-real time ETL reduces data latency compared to conventional ETL	Complexity in integration and system overhead	Develop lightweight frameworks for real-time analytics
Qu et al. (2015)	Snapshot maintenance in ETL flows	Incremental recompilation pipeline with query-driven policy	Efficient query-driven delta capture improves OLAP performance	Managing concurrent workloads and synchronization	Improve workload schedulers for better concurrency and latency control

6. Conclusion and Future Work

Decision support systems have always been a key component of data warehousing, and data warehousing has always been based on batch oriented ETL processes to integrate and prepare data. Though, the blistering development of big data and real-time applications made the focus on the streaming ETL pipelines, which allows organizations to process and analyze data on-the-fly instead of periodically. The shift of paradigm indicates the increasing need to be immediate, precise and scaleable in data-driven environments. The survey refers to the shift in conventional ETL systems to streaming-enabled systems, tools and techniques, such as Apache Kafka, Spark streaming, Apache Flink, and Apache NiFi. These architectures depict how the existing systems can handle continuous ingestion, ensure scalability, elasticity and fault tolerance and address the problem of data quality, consistency and integration.

The findings demonstrate that the streaming ETL pipelines can be applied to make organizations more responsive, business and efficient in their operations in various sectors like finance, healthcare and e-commerce. However, issues like overhead infrastructures, complexity of the systems, and incorporation of heterogeneous sources of data have existed, which constrain their smooth adoption. Future studies ought to cover lightweight architecture, smart automation applications and enhanced fault-tolerant systems to minimize overhead. Furthermore, the solution of security, interoperability and standardization will reinforce the use of streaming ETL in the industries, so as to have sustainable, scalable, and real-time adaptable data warehousing.

References

[1] K. Vassakis, E. Petrakis, and I. Kopanakis, “Big data analytics: Applications, prospects and challenges,” *Lect. Notes Data Eng. Commun. Technol.*, vol. 10, no. January, pp. 3–20, 2018, doi: 10.1007/978-3-319-67925-9_1.

[2] V. N. Gudivada, A. Apon, and J. Ding, “Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations,” *Int. J. Adv. Softw.*, vol. 10, no. 1, pp. 1–20, 2017.

[3] H. G. Kola, “Data Warehousing Solutions for Scalable Etl Pipelines,” *J. Sci. Res. Sci. Eng. Technol.*, vol. 4, no. 8, pp. 762–769, 2018.

[4] F. Xiao, C. Li, Z. Wu, and Y. Wu, “NMSTREAM: A scalable event-driven ETL framework for processing heterogeneous streaming data,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 4, no. 4, pp. 243–246, 2018, doi: 10.5194/isprs-annals-IV-4-243-2018.

[5] K. Kakish and T. a Kraft, “ETL Evolution for Real-Time Data Warehousing,” *Proc. Conf. Inf. Syst. Appl. Res.*, 2012.

[6] A. Simitsis, P. Vassiliadis, and T. Sellis, “Optimizing ETL Processes in Data Warehouses,” in *21st International Conference on Data Engineering (ICDE’05)*, IEEE, 2005, pp. 564–575. doi: 10.1109/ICDE.2005.103.

[7] S. Pillai and P. S. Metkewar, “Literature Review of Concerns Prevalent within Real Time Data Warehouse,” *Int. J. Trend Res. Dev.*, vol. 3, no. 4, pp. 370–371, 2016.

[8] N. R. Mandala, “The evolution of ETL architecture: From traditional data warehousing to real-time data integration,” *World J. Adv. Res. Rev.*, vol. 1, no. 3, pp. 073–084, Mar. 2019, doi: 10.30574/wjarr.2019.1.3.0033.

[9] J. Meehan *et al.*, “S-Store,” *Proc. VLDB Endow.*, vol. 8, no. 13, pp. 2134–2145, Sep. 2015, doi: 10.14778/2831360.2831367.

[10] V. A. Kherdekar and P. S. Metkewar, “A Technical Comprehensive Survey of ETL Tools,” *Int. J. Appl. Eng. Res.*, vol. 11, no. 04, Feb. 2016, doi: 10.37622/IJAER/11.4.2016.2557-2559.

[11] A. A. Yulianto, “Extract Transform Load (ETL) Process in Distributed Database Academic Data Warehouse,” *APTIKOM J. Comput. Sci. Inf. Technol.*, vol. 4, no. 2, pp. 61–68, Jul. 2019, doi: 10.11591/APTIKOM.J.CSIT.36.

[12] S. Gupta, N. Agrawal, and S. Gupta, “A Review on Search Engine Optimization: Basics,” *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 5, pp. 381–390, May 2016, doi: 10.14257/ijhit.2016.9.5.32.

[13] S. T and S. N. K, “A study on Modern Messaging Systems- Kafka, RabbitMQ and NATS Streaming,” *CoRR*

abs/1912.03715, 2019.
[14] A. Kushwaha, P. Pathak, and S. Gupta, "Review of optimize load balancing algorithms in cloud," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, pp. 1–9, 2016.
[15] M. Armbrust *et al.*, "Structured Streaming: A Declarative API for Real-Time Applications in Apache Spark," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 361–372, Oct. 2018, doi: 10.1145/1282427.1282421.
[16] R. Verma, "Real-Time Data Integration: The Next Evolution in ETL," *Int. Res. J. Eng. Technol.*, vol. 2, no. April, 2015, doi: 10.2139/ssrn.5000978.
[17] J. P. A. Runtuwene, I. R. H. T. Tangkawarow, C. T. M. Manoppo, and R. J. Salaki, "A Comparative Analysis of Extract, Transformation and Loading (ETL) Process," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 306, no. 1, p. 012066, Feb. 2018, doi: 10.1088/1757-899X/306/1/012066.
[18] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–52, Jul. 2009, doi: 10.1145/1541880.1541883.
[19] A. Pareek, B. Khaladkar, R. Sen, B. Onat, V. Nadimpalli, and M. Lakshminarayanan, "Real-time ETL in Striim," in *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics*, New York, NY, USA: ACM, Aug. 2018, pp. 1–10. doi: 10.1145/3242153.3242157.
[20] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 430–447, 2018, doi: 10.1109/TSC.2017.2711009.
[21] A. Sari and M. Akkaya, "Fault Tolerance Mechanisms in Distributed Systems," *Int. J. Commun. Netw. Syst. Sci.*, vol. 08, no. 12, pp. 471–482, 2015, doi: 10.4236/ijcns.2015.812042.
[22] N. R. Mandala, "Memory Management in Large-Scale ETL Processes," *Int. J. Nov. Res. Dev.*, vol. 2, no. 3, pp. 42–48, 2017.
[23] C. Batini *et al.*, "Data quality in remote sensing," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 42, no. 2W7, pp. 447–453, 2017, doi: 10.5194/isprs-archives-XLII-2-W7-447-2017.
[24] A. Simitsis, P. Vassiliadis, and T. Sellis, "Optimizing ETL Processes in Data Warehouses," in *21st International Conference on Data Engineering (ICDE'05)*, IEEE, 2005, pp. 564–575. doi: 10.1109/ICDE.2005.103.
[25] F. Alam and N. Kamal, "Survey on Data Warehouse from Traditional to Realtime and Society Impact of Real Time Data," *Int. J. Comput. Appl.*, vol. 177, no. 9, pp. 20–24, Oct. 2019, doi: 10.5120/ijca2019919463.
[26] A. Katari, "ETL for Real-Time Financial Analytics : Architectures and Challenges," *Innov. Comput. Sci. J.*, vol. 5, no. 1, pp. 1–17, 2019.
[27] H. Isah and F. Zulkernine, "A Scalable and Robust Framework for Data Stream Ingestion," *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 2900–2905, 2018, doi: 10.1109/BigData.2018.8622360.
[28] P. Chandra and M. K. Gupta, "Comprehensive survey on data warehousing research," *Int. J. Inf. Technol.*, vol. 10, no. 2, pp. 217–224, Jun. 2017, doi: 10.1007/s41870-017-0067-y.
[29] M. Minhaj, "An Exploratory Study of Near-Real Time ETL Approaches for the Design of Agile Business Intelligence Infrastructure Mohamed Minhaj," *SDM Res. Cent. Manag. Stud.*, vol. V, pp. 23–44, 2016.
[30] W. Qu, V. Basavaraj, S. Shankar, and S. Dessloch, "Real-Time Snapshot Maintenance with Incremental ETL Pipelines in Data Warehouses," in *Big Data Analytics and Knowledge Discovery*, S. Madria and T. Hara, Eds., Cham: Springer International Publishing, 2015, pp. 217–228.