Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 16, No. 2, 2026

Research on the Construction of an Automation Platform Based on Data Stream Processing

Lele Li

GRG METROLOGY & TEST (TIANJIN) CO., LTD. Tianjin, China

Abstract: With the deepening of digital transformation, to address the problems of high latency and low accuracy in processing industrial production automation data using traditional methods, this paper proposes a type of research on the construction of an automation platform based on data stream processing. The research is analyzed in a four-layer structure. Firstly, a machine learning method is adopted to perform zero-mean standardization on the accessed data; secondly, the stream processing engine layer is constructed using Apache Flink technology to process data streams continuously with low latency; then, the storage service layer of the platform is built based on metadata; finally, it is explained that the data operation, maintenance and application layer is responsible for the stable operation of the system and data output. The research results provide theoretical support and practical reference for enterprises to build an efficient, reliable, and easy-to-use real-time data automation platform.

Keywords: Data Stream Processing; Automation Platform; Machine Learning; Zero-mean; Flink.

1. Introduction

Data has become the core production factor of enterprises in the era of digital economy. Enterprise production data is characterized by high speed, large volume, heterogeneity and disorder, which are difficult to handle with traditional ETL and stand-alone systems. Stream processing technology achieves millisecond-level low latency by processing data efficiently as it is generated. Therefore, constructing an automation platform based on data stream processing has become an urgent demand for enterprise digital transformation.

Research on data stream processing has attracted the attention of experts from various industries. For example, Reference [1] proposes a decentralized cloud-edge-end collaborative data stream processing architecture based on service-oriented design to address the challenges of flexibility, scalability, and low latency in data stream processing under the background of big data. This architecture constructs an active data service model for large-scale data stream, abstracting data stream and its processing process into services with moderate granularity, independent deployment and dynamic scheduling, to realize the decoupling between data and computing. Furthermore, an event-driven mechanism is introduced, and an event-driven dynamic collaboration method for cloud-edge-end services is proposed, which significantly improves the flexible response capability of the system. Finally, experimental verification is carried out based on real power quality sensing data stream, and the results demonstrate the correctness and effectiveness of the proposed architecture. Reference [2] points out that with the rapid development of information technology, big data has gradually become a key resource in modern society. Among them, real-time data stream, as an important component of big data, has typical characteristics such as large data volume, high processing timeliness and relatively low value density. This research focuses on real-time data stream processing and analysis strategies based on big data, aiming to improve processing efficiency, deeply mine data value, and provide theoretical basis and technical reference for research and practice in related fields. Aiming at the problems of unbalanced data sets and poor resistance to concept drift, Reference [3] proposes a Tor traffic analysis model combining stacked denoising autoencoder (SDAE) and online sequential extreme learning machine (OS-ELM). Firstly, the original Tor traffic packets are segmented, denoised, and feature-extracted, the obtained one-dimensional feature sequences are converted into grayscale images, and then input into an improved multi-scale deep convolutional generative adversarial network to generate samples and achieve data set balance. Then, SDAE is used for sequence dimensionality reduction, and the extracted features are input into OS-ELM to realize online identification of Tor anonymous traffic. Finally, the experimental results show that the improved generative model can increase the model recognition rate by about 2.8 percentage points; the overall model accuracy reaches 95.7%, and the recognition efficiency is significantly better than traditional CNN and LSTM models. Thus, it can be seen that industrial automation production data based on data stream processing is more efficient than traditional models.

In summary, to solve the problems of high latency and low accuracy in processing industrial production automation data with traditional methods, this paper proposes a type of research on the construction of an automation platform based on data stream processing. Based on industrial data and adopting data stream processing technology, an automation platform capable of efficiently processing multi-source heterogeneous feature data is designed and built. The platform realizes the automation and visualization of the whole process of data access, processing, storage and service, effectively reducing the technical threshold of data stream processing. The research results provide theoretical basis and practical reference for enterprises to build an efficient, reliable and easy-to-use real-time data automation platform.

2. Related Theories and Technical Foundations

2.1. Data Stream Processing Technology

Data stream processing technology started in the 1990s and evolved from centralized to distributed. Early Storm had low

latency, lacked state management and consistency guarantee. Spark Streaming adopts micro-batch processing to balance latency and throughput; Flink, based on the native data stream model, supports event time, exactly-once processing, and state persistence, becoming the mainstream engine in the industry. Enterprises such as Alibaba Cloud have optimized Flink to support cloud-native and elastic scaling, improving resource utilization. Therefore, this paper adopts a Flink-based stream processing solution to build the automation platform [4].

2.2. Research Status of Data Automation Platforms

In terms of data governance of data automation platforms, the integration of AI-related technologies and data automation has become a new trend. Functions such as abnormal data detection and adaptive data pipeline optimization are realized through machine learning models. However, the overall integration of AI and automation processing is still not in-depth, and data quality-related problems still require manual intervention in most links. Therefore, this paper proposes an automation platform solution covering the whole process of data collection, processing, storage and governance, providing infrastructure support for building a more intelligent and efficient data governance mechanism.

2.3. Definition and Characteristics of Data Stream

Data stream in this paper refers to a continuously generated, unbounded, and uncertain-order data sequence, with characteristics such as continuity, heterogeneity and disorder. Continuity means that data from data sources is generated all the time without obvious start and end times; heterogeneity means that data sources have diverse formats, including structured data, semi-structured data and unstructured data; disorder means that the data arrival order may be inconsistent with the generation order, with out-of-order and delayed situations [5].

2.4. Data Stream Engine Technology

Apache Flink is one of the most widely used and mature stream processing engines at present, with core features of native stream processing, stream-batch integration, event time semantics and state management.

(1) Native stream processing is a computing rule based on the data stream programming model, which can process unbounded data streams continuously and in real time.

(2) Stream-batch integration refers to the unified processing of unbounded data stream and bounded batch data, with one set of core code supporting both stream processing and batch processing modes.

(3) Event time semantics means that the system can perform window calculation according to the actual occurrence time of events, and support the processing of out-of-order data caused by network delay and other reasons.

(4) State management means that the system provides a variety of state backends, which can persistently save states during calculation and support a fault-tolerant mechanism based on incremental snapshots.

3. Research Route

3.1. Overall Architecture Design

The research roadmap is divided into a four-layer structure from top to bottom, namely the data access layer, data stream engine layer, data storage layer, and data operation, maintenance, and application layer. The research roadmap of this paper is shown in Figure 1 below.

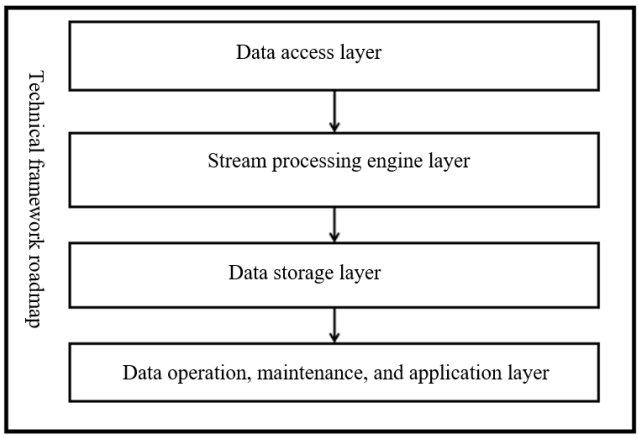

Figure 1. Research roadmap

3.2. Data Access Layer

The purpose of the data access layer is to realize the unified access and standardized processing of multi-source heterogeneous data, and a machine learning method is adopted to standardize the accessed data [6].

(1) Unified data access

Unified data access in this paper refers to collecting data from different sources and formats into a unified platform or system through standardized interfaces or protocols, realizing centralized management, unified scheduling and subsequent processing. The unified data access mode is as follows.

Suppose the original data source set is $D = \{D_1, D_2, \cdots, D_n\}$, each data source is D_i, and the goal of unified access is to convert it into a unified format F. The expression of unified access is shown in Formula (1).

$$D = \bigcup_{i=1}^{n} f_i(D_i), F = \{f_1, f_2, \cdots, f_i\} \tag{1}$$

Where D represents the data after unified formatting; F represents the function set, which aims to adopt different functions for different data sources and finally convert the data of multiple data sources into the same format.

(2) Data normalization

Because the data units of different equipment and systems in industrial production are different, they cannot be directly merged and processed. Therefore, based on Formula (1), data normalization is required to scale the original data proportionally to a specific interval or convert it into a distribution with a mean of 0 and a standard deviation of 1 [7]. This paper adopts zero-mean standardization, as shown in Formula (2).

$$D_j^{norm} = \frac{D_j - D_j^{\mu}}{D_j^{\sigma}} \tag{2}$$

Where D_j^{norm} represents the normalization result of the j-th column data, D_j^{μ} represents the mean of the j-th column data after normalization, and D_j^{σ} represents the variance of the mean of the j-th column data after normalization.

3.3. Stream Processing Engine Layer Design

The stream processing engine layer is built based on Apache Flink technology, serving as the core computing module of the entire platform, responsible for continuous and low-latency processing of data streams, mainly relying on its mature capabilities in state management, event time semantics and processing accuracy. It is mainly reflected in native real-time processing, state management and event time semantics, as detailed below [8-9].

(1) It supports native real-time processing of unbounded data. The difference from the micro-batch processing architecture is that Flink triggers calculation immediately when each data arrives instead of waiting for batch construction, thus controlling the processing latency to the millisecond level.

(2) In terms of state management, Flink supports RocksDB or memory as the state backend, and realizes state persistence and fault recovery combined with incremental checkpoints. When a task is abnormal, the system recovers from the latest checkpoint to ensure no data duplication or loss.

(3) As for event time semantics, Flink performs window aggregation according to the timestamp of the data itself, and processes out-of-order and late data with the help of the watermark mechanism. The window calculation result can be configured to output exactly once to prevent repeated triggering.

This engine layer uniformly consumes real-time streams from the data access layer, and after processing through operators such as filtering, aggregation and joining, outputs the results to downstream storage or alarm services.

3.4. Storage Service Layer Design

The storage service layer adopts metadata storage, and its core content includes basic descriptive information such as data source connection information and location paths, data table structure definitions and field types, data formats and encoding methods, data partitioning and indexing rules, data quality verification standards, data lineage records, data access control information, as well as data versions and timestamps.

Such metadata are uniformly stored in the metadata storage component for on-demand invocation by the data processing engine, query service and management module, to realize the organization, location, interpretation and control of physically stored data.

3.5. Data Operation, Maintenance, and Application Layer

The data operation, maintenance, and application layer is located at the top of the platform architecture, responsible for the stable operation of the system and data output.

(1) In terms of operation and maintenance, this layer provides functions such as job scheduling, task monitoring, abnormal alarm, log collection and resource management. It can track the running status, data throughput, processing latency and resource utilization of stream processing jobs in real time, and trigger automatic recovery or manual intervention when tasks fail or data is backlogged.

(2) In terms of application, this layer provides data query interfaces, message push and API gateways for business scenarios, supporting the output of processing results to relational databases, key-value stores or real-time dashboards. At the same time, this layer includes metadata management and data lineage display functions, facilitating operation and maintenance personnel and data users to view data sources, processing links and field mapping relationships. Through the unified operation and maintenance portal and application portal, the platform realizes centralized management and control of data tasks and convenient access to business results, which not only ensures the reliability of system operation, but also improves the availability and transparency of data services.

4. Summary and Outlook

This paper focuses on the construction of an automation platform for data stream processing. Aiming at the problems of insufficient real-time performance, low automation, and complex architecture in traditional data processing, a type of research on the construction of an automation platform based on data stream processing is proposed. This solution systematically combs the development context of data stream processing and automation technologies, deeply analyzes the principles and application scenarios of the Flink stream processing engine, and provides theoretical support for platform design.

However, this paper also has certain deficiencies. For example, the platform currently only supports simple machine learning model inference and lacks support for real-time deployment and optimization of deep learning models; the data governance function is not perfect, and intelligent capabilities such as automatic metadata identification and automatic data quality rule generation are insufficient. To address the above problems, this paper will deepen the research in the following two aspects in the future: firstly, realize intelligent functions such as automatic SQL generation, fault root cause analysis and automatic data quality rule optimization based on large language models. Secondly, cloud-edge-end collaboration: construct a cloud-edge-end collaborative data stream processing architecture, where edge nodes are responsible for data preprocessing and real-time analysis, and the cloud is responsible for global aggregation and long-term storage, meeting the needs of IoT edge computing scenarios.

References

[1] Zhang Shouli, Liu Chen. Research on Service-oriented Cloud-edge-end Collaborative Data Stream Processing Architecture [J]. Journal of Shandong Agricultural University (Natural Science Edition), 2024, 55(03): 385-395.

[2] Chen Juan. Research on Real-time Data Stream Processing and Analysis Strategy Based on Big Data [C]// China Technology Market Association, China High-Tech Industrialization Research Association, China International Association for Science and Technology Cooperation, Entrepreneurs Branch of China Future Research Association, Discover Magazine. Proceedings of the 23rd China Scientists Forum. Shanghai Cairongju Information Technology Co., Ltd.; 2024: 91-98. DOI: 10. 26914/c.cnkihy.2024.053511.

[3] Xi Rongkang, Cai Manchun, Lu Tianliang. Tor Traffic Analysis Model Based on Data Augmentation and Data Stream Processing [J]. Computer Engineering, 2023, 49(03): 177-184. DOI: 10.19678/j.issn.1000-3428.0064386.

[4] Y.Jing, W. Yafei and Z. Fan, Research and Application Strategy for Intelligent Car Platform Construction Based on Flink, Kafka Stream Data Technology and Deepseek, 2025 IEEE 3rd International Conference on Image Processing and

Computer Applications (ICIPCA), Shenyang, China, 2025, pp. 1-5, doi: 10.1109/ICIPCA65645.2025.11139039.

[5] X.Wang, J. Lu, F. Zhang and J. Yang, Automobile Brand Analysis System Based on Feature Engineering and Apache Kafka+Flink Stream Data Processing Framework, 2025 International Conference on Computer Science, Technology and Engineering (ICCSTE), Wuhan, China, 2025, pp. 128-133, doi: 10.1109/ICCSTE65902.2025.11138357.

[6] Wang, Y., Zhang, F., Feng, Q. et al. Strategic analysis of intelligent connected vehicle industry competitiveness: a comprehensive evaluation system integrating rough set theory and projection pursuit. Complex Intell.Syst. 10, 7033–7062 (2024). https://doi.org/10.1007/s40747-024-01525-w.

[7] Zhao Xiaoyu, Yang Xing, Bu Lei. Research on SQLIA Recognition Technology Combining Machine Learning and Feature Engineering [J]. Computer Programming Skills & Maintenance, 2025, (08): 129-132. DOI:10.16184/j. cnki. comprg. 2025.08.030.

[8] Kai, G. Yaxin and Z. Fan, Research on the Application of Flink Streaming Data Technology in the Construction of Automobile Internationalization Platform, 2025 Asia-Europe Conference on Cybersecurity, Internet of Things and Soft Computing (CITSC), Rimini, Italy, 2025, pp. 778-782, doi: 10.1109/ CIT SC64390.2025.00145.

[9] Y. Liu, Y. Wang and X. Yang, Research on the Construction of a Distributed Overseas Data Acquisition and Preprocessing Platform Based on the FLINK Real-Time Streaming Computing Framework, 2025 International Conference on Computer Science, Technology and Engineering (ICCSTE), Wuhan, China, 2025, pp. 01-07, doi: 10.1109/ ICCSTE 65902. 2025. 11138049.