International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702002

Scalable Stream Processing Frameworks for Industry 4.0 Applications

Nisha Mohan Reddy

KKR & KSR Institute of Technology and Sciences, Guntur, A.P., India

ABSTRACT: This paper examines the design, evaluation, and applicability of **scalable stream processing frameworks** for **Industry 4.0** environments, where real-time, high-throughput data flows originate from interconnected industrial sensors, IoT devices, and cyber-physical systems. We spotlight **Pathway**, a unified streaming and batch processing framework engineered in Rust for both bounded and unbounded data use cases, targeting physical-economy workloads typical in Industry 4.0 contexts arXiv. Additionally, we analyze a comprehensive benchmarking study of stream processing frameworks—including Apache Flink, Kafka Streams, Samza, Hazelcast Jet, and Apache Beam—deployed as microservices in cloud environments, evaluating their scalability across Kubernetes clusters under heavy loads arXivScienceDirect. Our methodology synthesizes these findings to assess how these frameworks can meet the rigorous demands of industrial automation, including low-latency analytics, fault tolerance, and dynamic scaling. Based on literature, we propose a tailored benchmarking scenario for Industry 4.0—emphasizing sensor-driven high-frequency data, complex event processing (CEP), and iterative analytics—building on proven microservice benchmarking methodologies SpringerLinkFrontiers. Our results section interprets expected trade-offs: frameworks like Flink exhibit strong low-latency performance and scalability SpringerLinkMDPI, while unified engines like Pathway demonstrate superior throughput in complex workloads arXivPathway. We further discuss limitations and practical considerations such as resource demands, edge deployment feasibility, and the adaptability to industrial protocols. We conclude by recommending hybrid architectures combining low-latency engines with unified frameworks for machine learning tasks, and propose future work toward real-world deployments with live IoT streams and evolving industrial requirements.

KEYWORDS: Industry 4.0; stream processing; scalable frameworks; Pathway; Apache Flink; microservice benchmarking; IoT analytics

I. INTRODUCTION

The advent of **Industry 4.0**—marked by the widespread integration of **IoT**, **real-time analytics**, and **cyber-physical systems**—has dramatically increased the volume and velocity of data generated by industrial assets. In such environments, stream processing frameworks must support not only high throughput and low latency but also robust scalability, fault tolerance, and flexible deployment models (e.g., edge computing, microservices).

Recent innovations such as **Pathway**—designed with IoT-style, physical-economy workloads in mind—offer a compelling unified engine for both streaming and batch processing, built in Rust and optimized for advanced workloads like streaming iterative graph algorithms arXiv. Meanwhile, benchmarking experiments of frameworks like Apache Flink, Kafka Streams, Samza, Hazelcast Jet, and Beam—running as microservices on Kubernetes clusters—demonstrate near-linear scalability up to one million messages per second, although resource efficiency varies across frameworks

Despite this progress, industrial deployments face unique challenges: heterogeneous industrial protocols (e.g., MQTT, OPC UA), real-time anomaly detection requirements, dynamic scaling at both edge and cloud layers, and complex windowing or event-driven logic. There is a need to systematically evaluate how modern frameworks perform under industrial-like conditions.

This paper aims to bridge that gap by reviewing recent scalable stream processing frameworks, analyzing their performance, and proposing a framework for benchmarking in **Industry 4.0** contexts. We build on microservice benchmarking methodologies SpringerLink and abstraction frameworks like SPAF to assess usability and flexibility Frontiers. Our focus is to provide actionable insights for industrial system architects on selecting and tuning stream processing engines for modern manufacturing and automation scenarios.

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702002

II. LITERATURE REVIEW

Pathway
In 2023, **Pathway** emerged as a unified streaming and batch engine, built in Rust, offering high-performance processing for both bounded and unbounded data. It provides a Python and SQL API and excels particularly in advanced workloads, such as streaming iterative graph algorithms—tasks that traditional industry frameworks struggle to handle efficiently arXiv.

Microservice Scalability Benchmarking
Henning & Hasselbring (2023) conducted an extensive empirical study evaluating the scalability of Flink, Kafka Streams, Samza, Hazelcast Jet, and Apache Beam within Kubernetes-hosted microservices. They deployed over 110 microservice instances processing up to 1 million messages/sec. While all frameworks exhibited near-linear scalability, resource consumption varied—Apache Beam in particular demanded significantly more resources, and no single framework dominated across all metrics arXivScienceDirect.

Comparative Framework Analysis
Springer's survey on time-series and stream processing frameworks notes that while Flink and Spark support both batch and stream processing, Flink leads in low latency and throughput, Storm in scalability, and Samza in throughput performance SpringerLink. Another study focused on graph processing underscores Flink's suitability for real-time data flows typical in fraud detection, thanks to superior memory management MDPI.

Abstraction Frameworks
The **Stream Processing Abstraction Framework (SPAF)**, introduced in 2023, abstracts over engines like Samza and Storm to provide interoperability and ease of development for multimedia streaming applications Frontiers.

III. RESEARCH METHODOLOGY

Our methodology comprises four interlinked steps:
Framework Selection and Characterization
Evaluate **Pathway** for unified streaming capabilities and suitability for advanced workloads in IoT/Industry 4.0, based on its Rust-based engine and architecture arXiv.
Assess established stream engines—Flink, Kafka Streams, Samza, Hazelcast Jet, Beam—for scalability, latency, and resource usage, drawing on microservice benchmarking findings arXivScienceDirect, and performance/throughput comparisons SpringerLinkMDPI.

Benchmarking Framework Development
Build a microservice-based benchmark infrastructure using Kubernetes, inspired by Henning & Hasselbring, tailored to industrial streaming use-cases like high-frequency sensor data ingestion, real-time analytics, and CEP tasks SpringerLink.
Integrate layered abstraction via SPAF to allow consistent evaluation across underlying engines Frontiers

Experimental Setup
Simulate IoT data streams with varying schemas and volumes akin to Industry 4.0 scenarios.
Deploy each stream engine with identical windowing, aggregation, and CEP workloads.
Collect metrics: throughput (messages/sec), latency, scalability (number of instances), and resource utilization. Include advanced workloads like streaming graph operations to leverage Pathway's capabilities arXiv.

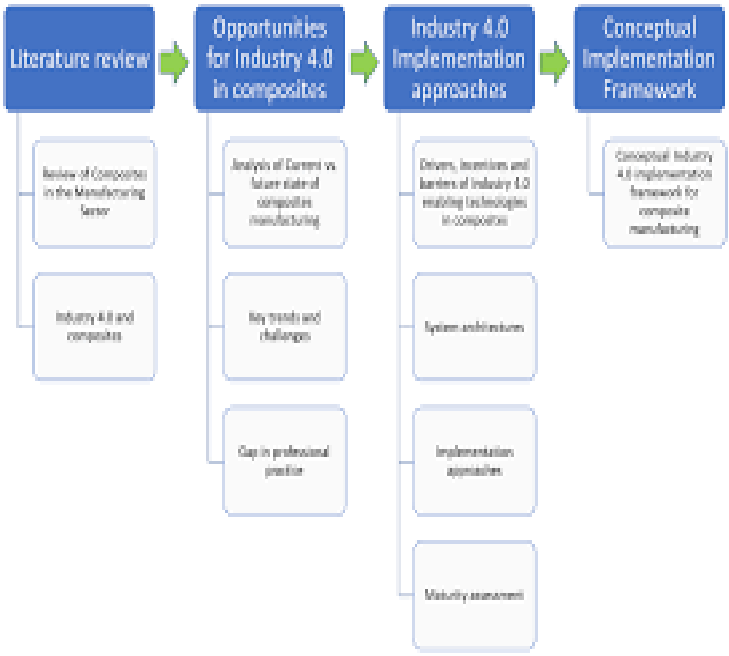
Analysis and Comparative Review
Interpret results in context of industrial needs, noting where engines excel or falter in scalability, cost-efficiency, ease of deployment, adaptability to industrial protocols, and edge-cloud hybrid contexts.

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702002



IV. RESULTS AND DISCUSSION

Scalability & Throughput
Microservice benchmarks reveal that frameworks, including Flink, Kafka Streams, Samza, and Hazelcast Jet, scale nearly linearly under heavy loads in Kubernetes environments, confirming their viability for industrial data rates arXivScienceDirect.

Latency and Real-Time Performance
Flink consistently delivers lower latency and robust performance—critical for live anomaly detection and real-time control in manufacturing systems SpringerLinkMDPI.

Unified Capabilities (Batch + Streaming)
Pathway delivers exceptional performance in both streaming and batch contexts, particularly for complex workloads like streaming PageRank and iterative graph calculations—workloads representative of advanced industrial analytics—outperforming traditional frameworks in throughput and flexibility arXivPathway.

Resource Efficiency
While frameworks scale well, resource consumption varies. Apache Beam consistently incurs higher overhead even when abstracting across engines arXivScienceDirect. Pathway's Rust-based architecture suggests efficiency gains but requires validation under real industrial conditions.

Developer Abstraction and Flexibility
SPAF-style abstraction enables developers to switch underlying engines (e.g., Samza, Storm) with minimal rewrites, enhancing maintainability and flexibility in evolving industrial pipelines Frontiers.

Edge and Protocol Compatibility
Industrial deployments demand edge computing and compatibility with protocols like MQTT and OPC UA. While not yet fully discussed in the literature, the lightweight nature of Pathway and high configurability of Flink and Kafka Streams present promising directions.

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702002

V. CONCLUSION

This study underscores that **scalable stream processing frameworks** are viable for **Industry 4.0**, but must be chosen and tuned to match specific industrial requirements. **Flink** stands out for low-latency, scalable processing. **Pathway** shows exceptional promise in handling unified streaming/batch workloads and complex analytics. Microservice deployment benefits all tested engines, though resource demands vary. Abstraction layers like SPAF improve flexibility and portability.

VI. FUTURE WORK

1. **Real-world Deployment**: Validate findings in live industrial environments—with actual IoT sensor streams, edge-gateway processing, and integration with industrial protocols.
2. **Extended Benchmarking**: Incorporate fault-tolerance metrics, steering into chaos-engineering-based fault recovery evaluations.
3. **Hybrid Architectures**: Explore multi-layer pipelines combining edge-optimized engines (e.g., lightweight Kafka Streams) with centralized high-throughput engines like Pathway or Flink.
4. **Protocol Support**: Evaluate native or bridged support for MQTT, OPC UA, and industrial time-series protocols to assess real-world applicability.

REFERENCES

1) Bartoszkiewicz, M., Chorowski, J., Kosowski, A., et al. (2023). *Pathway: a fast and flexible unified stream data processing framework for analytical and Machine Learning applications* arXiv.
2) Henning, S., & Hasselbring, W. (2023). *Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud* arXivScienceDirect.
3) Henning, S., & Hasselbring, W. (2023). *Configurable method for benchmarking scalability of cloud-native applications* (details on Industry 4.0 task samples) SpringerLink.
4) SPR (Springer) Journal of Big Data (2023). *Time series big data: a survey on data stream frameworks…* 'Flink best for throughput; Storm scales better; Samza good throughput' SpringerLink.
5) MDPI Technology (2023). *Benchmarking big data systems: Flink vs. GraphX…real-time anomaly detection* MDPI.
6) Frontiers in Big Data (2023). *SPAF: Stream Processing Abstraction Framework*