Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(5):45-50

Research Article ISSN: 2394 - 658X

Apache Kafka vs. Amazon Kinesis: A Detailed Comparison of Streaming Data Platforms for Real-Time Data Processing

Girish Ganachari[1], Rameshbabu Lakshmanasamy[2]

Email: girish.gie@gmail.com

ABSTRACT

This paper compares the architecture, performance, and scalability of Apache Kafka and Amazon Kinesis, as well as identifies whether using the latter can be more cost-efficient for real-time data processing. The aim of the study is to assist organisations in identifying the most suitable platform depending on the organisation needs and the conditions under which it will be operating.

Keywords: Apache Kafka, Amazon Kinesis, real-time data processing, streaming data platforms, performance evaluation, scalability, cost-efficiency, hybrid models, machine learning integration, data security, cross-platform interoperability.

INTRODUCTION

The introduction compares Apache Kafka and Amazon Kinesis in terms of their designs, performance, scalability, and cost implications in real-time data processing. The idea is to help organizations choose the most suitable platform for real-time data processing.

AIM AND OBEVTIVES

This study aims to carry out a performance comparison of Apache Kafka and Amazon Kinesis to determine their utility in real-time data processing, and their robustness in terms of capacity and price.

- To assess the basic functionalities of Apache Kafka and Amazon Kinesis.
- To compare the speed and efficiency of communication of both platforms in processing real-time data.
- To compare the aspects of Apache Kafka with the aspects of Amazon Kinesis to determine the scalability features supported by both platforms.
- To ascertain the cost implications of implementing as well as running the two platforms.

LITERATURE REVIEW

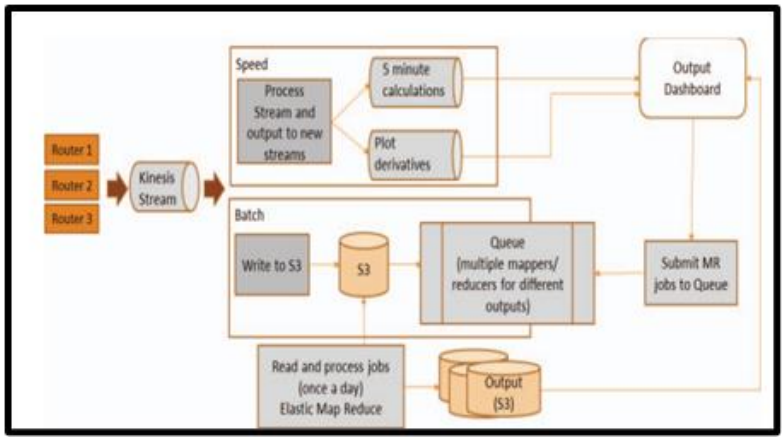

Fig. 1: *Main lambda architecture implemented on Amazon web services.*

The literature review discusses the basic components and operations of Apache Kafka and Amazon Kinesis from the context of real-time data processing. Apache Kafka is known for its highly distributed design successfully handling large data streams with the help of brokers, topics, and partitions [1]. Research must underline the suitability of map-reduce for situations that call for massive amounts of data to be transmitted and involve little lag time. Kinesis from Amazon is known to be easy to add AWS services and accommodate more traffic through shard operations and efficient real-time data analysis and log running. Such comparative evaluations substantiated Kafka's accuracy in handling complex events and Kinesis' advantage in terms of usability and serverless operations [2]. Especially in previous work, the focus is on what has been defined as the three main criteria for choosing a streaming data platform, namely performance, scalability, and cost. However, there remain some issues of understanding complicated interrelationships across these systems in different real-time data processing scenarios, which indicates the importance of further comparative research.

BENEFITS

A. Apache Kafka

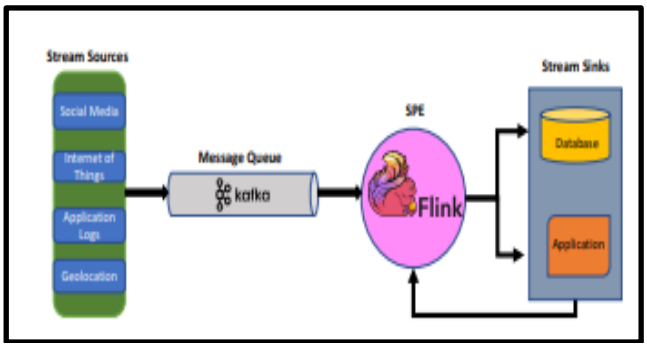

Fig. 2: *High-level view of a streaming data processing Pipeline*

Kafka possesses high throughput performance, which makes it suitable for handling large amounts of messages and doing it promptly. The stability of the system due to the robustness of the architecture ensures that errors can be handled and the core responsibility of reliability can be provided, thus making it ideal for critical and vital tasks [3]. The representational nature of Kafka makes horizontal scaling possible, especially through the introduction of more brokers and partitions for seamless handling of more data.

B. Amazon Kinesis

Kinesis' usability is excellent because it combines well with other AWS services, which is valuable for those utilizing AWS technologies. The multidimensional scaling mechanism of the system by the shard allows for the stream management of data, while not bogging down the operations tasks [4]. Real-time analysis is also well handled in Kinesis making it useful for such tasks as the ingestion of log and event data.

C. Comparative Analysis

They increase the efficiency of operations and enable real-time analysis on both platforms. However, the choice is still made between them depending on the specific purposes, available infrastructure, and price level.

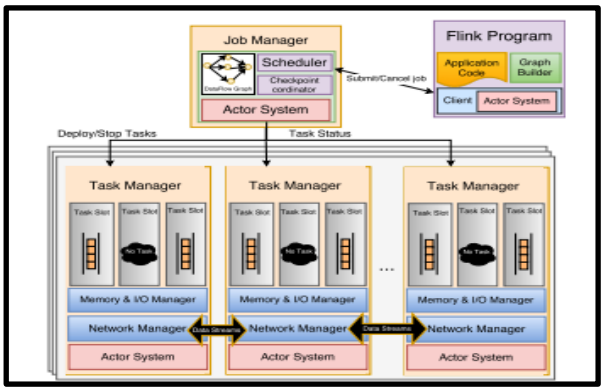

Fig. 3: *Architectural Overview of Apache Flink*

Real-time analytics, event sourcing, and stream processing have been incorporating Kafka largely. In the reliability context of financial services, it helps determine cases of fraud and/or monitor transactional activities through proper analysis of high transactional datasets [5]. This makes Kafka fit for integration in the health sector because it allows

real-time showing and analysis of patient data from multiple sources. Also, e-commercial platforms use Kafka for inventory management purposes and for keeping customers engaged with real-time streaming.

B. Amazon Kinesis

Kinesis is primarily used for the consumer of log and event information, which is suited for monitoring and analyzing IT problems. Its cooperation with AWS services enriches this technology in serverless contexts and makes it possible to use in marketing, for example, real-time clickstream analysis, and operative logistics intelligence [6]. Kinesis proves particularly useful when it comes to live video streaming in the media industry because its function and purpose is to present real-time information that contributes to the absence of disruptions in the user experience.

C. Comparative Analysis:

They are both useful for applications such as data processing in real-time; however, depending on the specific need of the business and the existing architecture, one will suffice for the other.

METHODOLOGIES

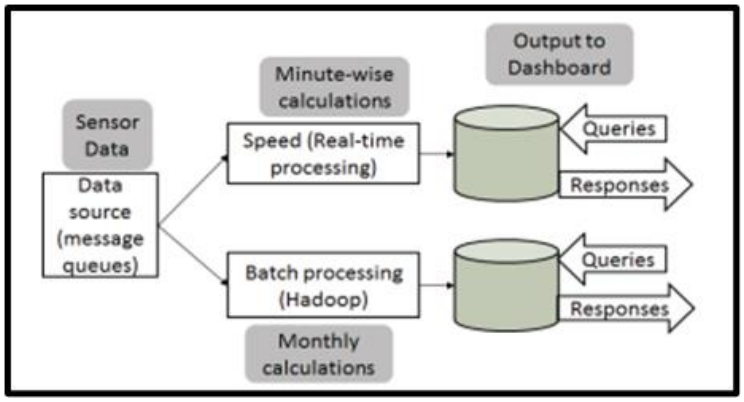

Fig. 4*: Basic lambda architecture for speed and batch processing*

The comparative analysis utilized in this study anchors on the methodological approaches that adopt secondary data from articles, case studies, and technical reports. The study comprises a systematic review of existing literature concerning Apache Kafka and Amazon Kinesis, more so in terms of their structure, performance, scalability, and costs [7]. This data is obtained from several papers outlining different industries and research papers to help explain the relevance and demerits of all reaches [8]. Thus, depending on the values achieved and comparing it with the comparative parameters such as bandwidth, response time, reliability of the system as well as the cost factor, one can have more meaningful and tangible results while considering real-time data processing for other applications.

RESULTS AND DISCUSSION

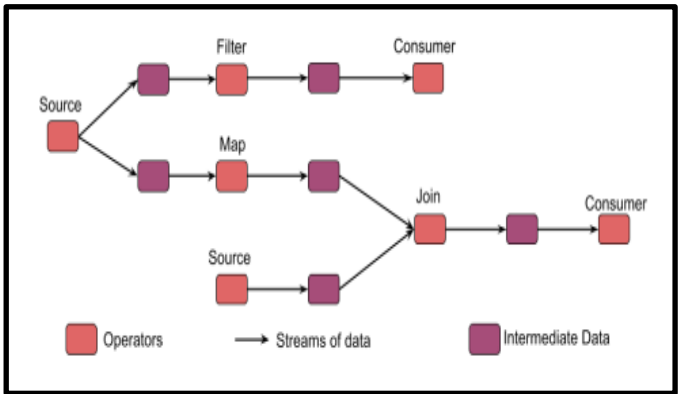

Fig. 5: *A Simple Dataflow Model*

Apache Kafka has otherwise been proven to be effective in real-time data processing especially in big data as has been witnessed when compared with Amazon Kinesis.

A. Performance

Apache Kafka offers some of the most encouraging outcome in terms of its throughput with little or no latency for abundant information. This makes it suited for scenarios where data ingestion and analyst need to happen with a relatively high velocity [9]. The study reveals that Kafka has high efficiency as it can handle millions of messages

in a second and has low latency; for this reason, Kafka should be implemented in applications that require real-time analytics and event sourcing [10]. However, it is also important to note that Amazon Kinesis is also a system that has excellent performance throughout the AWS family.

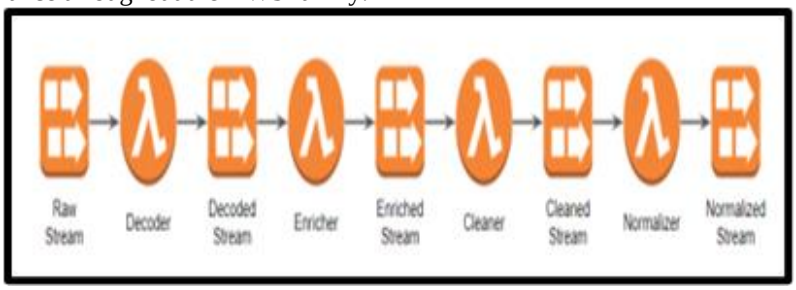

***Fig. 6**: Current Data Pre-processing Pipeline*

B. Scalability

As with Kafka having great scalability, the same goes for Kinesis even though they use somewhat different approaches. Therefore, Kafka improves the amount of brokers and partitions horizontally and increases its abilities of the effective data workload consumption [11]. In particular, horizontal scaling is effective for large-scale applications that require high availability and the ability to increase processing speed when data volumes increase [12]. Given that Kinesis can scale up with Shards, it is flexible in that the capacity for handling data flow can be easily adjusted.

C. Cost-Efficiency

Affordability is one of the crucial aspects to bear in mind when selecting a streaming data platform. Kafka is regarded more as an on-premise or a self-hosted solution and Maybe that's why it's cheaper for organizations that are capable of management at their level [13]. The costs of Kafka enterprise mainly pertain to the overhead costs that are related to its infrastructure and operations [14]. On the other hand, Kinesis, because it is fully managed, is easier to use with less operational overhead, but this may come at a higher cost that depends on usage patterns and the amount of data that needs to be processed.

D. Use Cases and Industry Applications

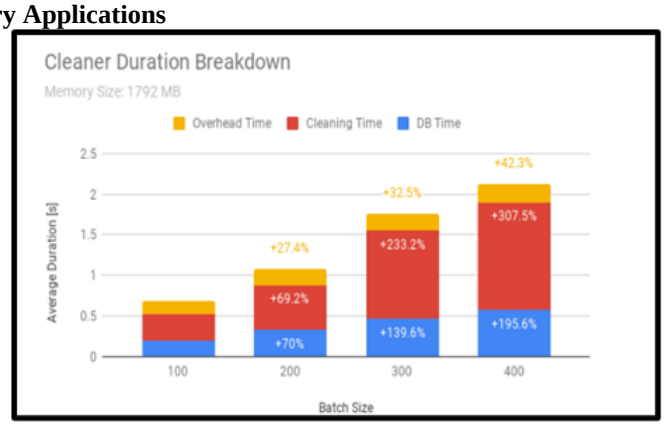

***Fig. 7:** Cleaner Duration Breakdown*

Kafka is actively used where it is necessary to process millions of records without the need for deep managing events, or in the financial and e-commerce industries [15]. One of the greatest strengths of this technology is the ability to work with a variety of big data applications and architectures, to build data pipelines at scale [16]. Kinesis is highly preferred in cloud-native architectures for use cases like IT monitoring, log data ingestion, and real-time analytics, because it fits perfectly in the AWS environment.

PERFORMANCE EVALUATION

The comparison of Apache Kafka and Amazon Kinesis in the scenario points out considerable benefits in the context of processing real-time data. In high traffic Apache Kafka has very good peak throughput and low latency, handling millions of messages per second [17]. The mentioned structure of the system ensures high efficiency and stability regardless of the data complexity. Amazon Kinesis stands as a variance contained within the AWS ecosystem that offers high-performance elements that can be expanded in proportion to shards [18]. However, Kinesis shows slightly more latency as compared to Kafka, especially in complex stream processing scenarios [19]. As for the two solutions, both of them are provided with the means of ensuring efficiency even with larger data sets.

CONCLUSION

The comparative analysis of the highly efficient platforms states that Apache Kafka and Amazon Kinesis are both suitable for processing real-time data. Kafka stands out as the best when it comes to online processing of massive volumes of data with little while Kinesis has the integration capability with AWS solutions. Still, while deciding between the two, the specifics of the use case and infrastructure, as well as the costs, should be considered.

FUTURE WORK

A. Hybrid Approaches

A further cross-sectional study could be done to compile the strengths of Apache Kafka with those of Amazon Kinesis [20]. These methods might use Kafka for the initial data-taking capacity that is high with the help of Kafka, and Kinesis for better integration with the AWS further for processing and observation of the data [21]. This would increase efficiency and profitability, which are two essential ingredients to financial sustainability [22].

B. Advanced Machine Learning Integration

Real-time data streams can be enriched through integrating state-of-art machine learning models; thus enhancing prediction and decision-making processes [23]. Future investigations should focus on designing and deploying ML algorithms that seamlessly operate with Kafka and Kinesis, increasing real-time data analysis [24].

C. Enhanced Security Measures

Future work in this area should shift to the enhancement of the security standards of both Kafka and Kinesis with an emphasis on the protection of data privacy [25]. This also means adopting new advanced encryption methods, strengthening access control, and following the roll changes for regulations on data confidentiality.

REFERENCES

[1]. Pampliega, J.M., 2016, November. Towards an architecture for real-time event processing. In II Simposio Argentino de GRANdes DAtos (AGRANDA 2016)-JAIIO 45 (Tres de Febrero, 2016).

[2]. Zhang, P., 2019. Performance Analysis of Cloud-Based Stream Processing Pipelines for Real-Time Vehicle Data.

[3]. Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I. and Zaharia, M., 2018, May. Structured streaming: A declarative api for real-time applications in apache spark. In Proceedings of the 2018 International Conference on Management of Data (pp. 601-613).

[4]. Kiran, M., Murphy, P., Monga, I., Dugan, J. and Baveja, S.S., 2015, October. Lambda architecture for cost-effective batch and speed big data processing. In 2015 IEEE international conference on big data (big data) (pp. 2785-2792). IEEE.

[5]. Javed, M.H., Lu, X. and Panda, D.K., 2017, December. Characterization of big data stream processing pipeline: A case study using flink and kafka. In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 1-10).

[6]. Terzi, D.S., Demirezen, U. and Sagiroglu, S., 2016. Evaluations of big data processing. Services Transactions on Big Data, 3(1), pp.44-53.

[7]. Falk, E., Gurbani, V.K. and State, R., 2017. Query-able kafka: An agile data analytics pipeline for mobile wireless networks. Proceedings of the VLDB Endowment, 10(12), pp.1646-1657.

[8]. Goldin, E., Feldman, D., Georgoulas, G., Castaño, M. and Nikolakopoulos, G., 2017, July. Cloud computing for big data analytics in the Process Control Industry. In 2017 25th Mediterranean Conference on Control and Automation (MED) (pp. 1373-1378). IEEE.

[9]. Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F. and Khan, S., 2019. A survey of distributed data stream processing frameworks. IEEE Access, 7, pp.154300-154316.

[10]. Ounacer, S., Talhaoui, M.A., Ardchir, S., Daif, A. and Azouazi, M., 2017. A new architecture for real time data stream processing. International Journal of Advanced Computer Science and Applications, 8(11).

[11]. Kılınç, D., 2019. A spark-based big data analysis framework for real-time sentiment prediction on streaming data. Software: Practice and Experience, 49(9), pp.1352-1364.

[12]. Teixeira, I.C.F., 2019. Event-Driven Real-Time Streaming Approach for Big Data, applied to an End-to-End Supply Chain.

[13]. Zdravevski, E., Lameski, P., Dimitrievski, A., Grzegorowski, M. and Apanowicz, C., 2019, December. Cluster-size optimization within a cloud-based ETL framework for Big Data. In 2019 IEEE international conference on big data (Big Data) (pp. 3754-3763). IEEE.

[14]. Corrao, S., 2019. Near real-time clickstream analysis: a journey in big data systems and architectures.

[15]. Sharvari, T. and Sowmya Nag, K., 2019. A study on modern messaging systems-kafka, rabbitmq and nats streaming. CoRR abs/1912.03715.

[16]. Elsner, J., Sivicki, T., Meisen, P., Meisen, T. and Jeschke, S., 2016. Implementing a Volunteer Notification System Into a Scalable, Analytical Realtime Data Processing Environment. Automation, Communication and Cybernetics in Science and Engineering 2015/2016, pp.841-853.

[17]. Córdova, P., 2015. Analysis of real time stream processing systems considering latency. University of Toronto patricio@ cs. toronto. Edu.
[18]. Taher, N.C., ImaneMallata, N.A. and Mawassc, N., IoT, Cloud Computing and Big Data Techniques for Smarter Healthcare.
[19]. Barba-González, C., Nebro, A.J., Benítez-Hidalgo, A., García-Nieto, J. and Aldana-Montes, J.F., 2020. On the design of a framework integrating an optimization engine with streaming technologies. Future Generation Computer Systems, 107, pp.538-550.
[20]. Persico, V., Pescapé, A., Picariello, A. and Sperlí, G., 2018. Benchmarking big data architectures for social networks data processing using public cloud platforms. Future Generation Computer Systems, 89, pp.98-109.
[21]. Sahal, R., Breslin, J.G. and Ali, M.I., 2020. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of manufacturing systems, 54, pp.138-151.
[22]. Jayaraman, P.P., Perera, C., Georgakopoulos, D., Dustdar, S., Thakker, D. and Ranjan, R., 2017. Analytics-as-a-service in a multi-cloud environment through semantically-enabled hierarchical data processing. Software: Practice and Experience, 47(8), pp.1139-1156.
[23]. Saif, S. and Wazir, S., 2018. Performance analysis of big data and cloud computing techniques: a survey. Procedia computer science, 132, pp.118-127.
[24]. Barika, M., Garg, S., Chan, A., Calheiros, R.N. and Ranjan, R., 2019. IoTSim-Stream: Modelling stream graph application in cloud simulation. Future Generation Computer Systems, 99, pp.86-105.
[25]. Imai, S., Patterson, S. and Varela, C.A., 2017, May. Maximum sustainable throughput prediction for data stream processing over public clouds. In 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (pp. 504-513). IEEE.