The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2023

Volume 24, Pages 177-183

IConTech 2023: International Conference on Technology

Utilizing Flink and Kafka Technologies for Real-Time Data Processing: A Case Study

Alper Bozkurt
Turkcell Odeme ve Elektronik Para Hizmetleri A.S.
(Paycell Research and Development Center)

Furkan Ekici
Atmosware Teknoloji Egitim ve Danısmanlık A.S.

Hatice Yetiskul
Turkcell Odeme ve Elektronik Para Hizmetleri A.S.
(Paycell Research and Development Center)

Abstract: In today's very competitive business world, being able to use data to its fullest in real time has become a key differentiation. This paper looks at how two cutting-edge technologies, Apache Flink and Apache Kafka, work together and how they are changing the way real-time data is processed and analyzed. With its fault-tolerant framework made for collecting data from many sources, Apache Kafka is a leader in reliability and scalability when it comes to ingesting data. Apache Flink is the perfect partner for Kafka because it is great at stream processing and low-latency event handling. This paper carefully explains how these technologies work together to create a complete set of tools for handling and analyzing data in real time. The paper goes into detail about how Flink and Kafka can work together, showing how data streams can be handled and intelligently put together to produce insights that can be used. This set of tools, which was created after a lot of study and real-world experience, helps organizations that want to start using real-time data in new ways. Evaluations of performance, scalability, and real-world applications show that this integrated method has a real effect. Beyond just talking about ideas, this study paper gives organizations a step-by-step plan for how to use real-time data to improve their decision-making. By taking advantage of how well Flink and Kafka work together, companies can become more flexible, quick to respond, and creative.

Keywords: Complex event processing, Event management systems, Real-time data management, Streaming data

Introduction

In today’s business landscape, companies aspire to gain a competitive edge by extracting value from vast streams of data in a rapid manner. In this context, real-time data processing and analysis are becoming increasingly essential. This paper will focus on how Apache Flink and Apache Kafka, both real-time data processing technologies, can be effectively utilized together and highlight scenarios where they excel.

This research paper embarks on an exploration of the dynamic synergy between two cutting-edge technologies: Apache Flink and Apache Kafka. These technologies, individually formidable, converge to create a potent ecosystem that empowers enterprises to seamlessly process, analyze, and act upon streaming data in real time. By delving into the intricacies of their integration and unveiling the nuances of effective data stream management, we uncover a realm of opportunities that promise to revolutionize how organizations leverage their data assets.

As the data deluge intensifies, Apache Kafka emerges as a beacon of reliability and scalability in data ingestion. Its architecture, built around fault tolerance and distributed commit logs, forms a robust foundation for aggregating data from myriad sources. Meanwhile, Apache Flink, with its prowess in stream processing and low-latency event handling, presents itself as the ideal counterpart. This paper unravels the seamless fusion of these technologies, illuminating the path toward constructing a comprehensive toolkit for real-time data analysis and processing.

In the subsequent sections, we traverse the landscape of Flink and Kafka integration, illustrating how data streams are not only corralled but also intelligently orchestrated to drive actionable insights. Our toolkit, a product of meticulous research and practical insights, is poised to become a guiding compass for organizations embarking on the journey of realtime data exploitation. Additionally, a thorough evaluation of performance, scalability, and applications showcases the tangible impact of our integrated approach in diverse real-world scenarios.

In essence, this research paper navigates beyond the realm of theoretical discourse. It is a tangible blueprint for organizations aspiring to fortify their decision-making capabilities with real-time data prowess. By embracing the symbiotic relationship between Flink and Kafka, businesses can chart a course toward unparalleled agility, responsiveness, and innovation in an era where data is the ultimate currency.

The organization of the paper is as follows: Literature Review: Key concepts in stream processing, Apache Flink and Apache Kafka architectures and features, and case studies of using Apache Flink and Apache Kafka in real-world applications. Methodology: Business process and development methodology to be followed in the project, key steps in project implementation, and evaluation criteria for the prototype application. Prototype Application and Evaluation of Application; overview of the prototype application, evaluation results and analysis, and discussion of limitations and future work. Conclusion: Summary of key points and recommendations for future research and development.

Literature Review

Apache Flink and Apache Kafka stand as pillars of modern data processing and streaming architecture, each bringing distinct capabilities that, when harnessed together, unlock a new realm of possibilities in real-time data analysis

Apache Flink: Empowering Real-Time Stream Processing: Apache Flink, a state-of-the-art stream processing framework, redefines the landscape of real-time data processing with its advanced features and flexibility (Apache Flink, 2023). At its core, Flink enables the processing of unbounded streams of data with low latency and high throughput. This unique trait positions it as a fundamental technology for applications demanding realtime insights. Flink’s inherent support for event time processing, coupled with its windowing capabilities, empowers developers to extract meaningful context from streams of data, regardless of the order in which events arrive. This ensures accurate computations for time-sensitive scenarios such as financial analytics, IoT data aggregation, and dynamic market analysis. Moreover, Flink’s stateful processing capabilities introduce the ability to maintain and update state across data streams, enabling complex event-driven workflows and pattern detection. Its rich ecosystem of connectors and libraries facilitates seamless integration with various data sources, sinks, and external systems, cementing its role as a versatile and robust tool for real-time stream processing.

Apache Kafka: The Foundation of Data Streaming: In the realm of data streaming and ingestion, Apache Kafka emerges as a cornerstone technology that addresses the challenges of data movement, reliability, and scalability (Apache Kafka, 2023). Kafka’s design revolves around a distributed publish-subscribe architecture, where data producers publish records to specific topics, and consumers subscribe to these topics to access the data. Kafka's commit log-based storage model guarantees durability and fault tolerance, ensuring that no data is lost even in the face of hardware failures. This makes Kafka a trusted and resilient platform for data ingestion, consolidation, and distribution across a multitude of applications. Its horizontal scalability and ability to handle massive volumes of data position it as a linchpin for high-throughput streaming pipelines. Furthermore, Kafka's integration with various data systems, including batch processing frameworks, databases, and analytics tools, makes it an ideal hub for ingesting data from diverse sources and routing it to the appropriate destinations, all while maintaining data integrity and consistency. Kafka Streams, a relentless stream processing client lurking within the Kafka abyss, enforces a bleak paradigm of read-process-write cycles, where every tortured processing

state change and the haunting echoes of result outputs are mercilessly etched into the unending annals of the Kafka log, like a sinister pact with an unforgiving abyss (Wang et al., 2021).

The Convergence: Flink and Kafka in Harmony: When Apache Flink and Apache Kafka converge, they form a symbiotic relationship that caters to the holistic needs of real-time data processing. Kafka's role as a reliable data ingestion and distribution platform seamlessly aligns with Flink's prowess in processing and analyzing data streams. The Kafka-Flink integration facilitates a continuous flow of data from Kafka topics to Flink streams, creating a dynamic pipeline that supports intricate event processing, real-time analytics, and complex computations. This convergence allows organizations to capitalize on the strengths of both technologies, offering a comprehensive solution that handles the entire lifecycle of real-time data—from ingestion and processing to analysis and action. It is this harmonious partnership that serves as the foundation for the toolkit and exploration presented in this research paper, ushering in a new era of rapid, intelligent, and data-driven decision-making.

Kafka and Flink: Stream Processing and High-Performance Data Processing Tools: In today's world, big data processing and stream processing applications play a crucial role in managing and analyzing rapidly growing data streams. Two significant tools that fulfill these needs are Apache Kafka and Apache Flink.

Kafka: Managing and Distributing Data Streams: Apache Kafka is an open-source stream processing platform designed to manage and distribute data streams reliably (Wiatr et al., 2018). Its key features include high durability, scalability, and real-time data stream processing capabilities. These features make Kafka suitable for various application scenarios. A study conducted by Wiatr et al. (2018) explored how Kafka can be optimized, especially in systems requiring low latency for data processing. This study provides valuable insights for organizations aiming to enhance Kafka's performance and process data streams more efficiently. Kafka's durability and fault tolerance mechanisms make it a robust choice for handling mission-critical data streams. Its distributed architecture allows for horizontal scaling, ensuring that Kafka can handle large volumes of data seamlessly. Additionally, Kafka's real-time data processing capabilities enable organizations to react to events as they occur, making it suitable for use cases like fraud detection, monitoring, and real-time analytics.

Flink: Combining Stream and Batch Data Processing: Apache Flink is a data processing platform that combines stream and batch data processing functionality (Carbone et al., 2015). Flink is designed to process data in real-time while also supporting batch data processing operations. This versatility makes Flink a powerful tool. A paper presented by Carbone et al. (2015) details the design and usage of Apache Flink. Flink's data processing capabilities offer various advantages to users in large data processing projects. Flink's unified processing model allows organizations to build both real-time and batch processing applications using a single framework. This reduces the complexity of managing multiple systems and simplifies the development process. Furthermore, Flink's support for event time processing ensures accurate handling of time-sensitive data, which is crucial for use cases such as financial analytics and IoT data processing.

The primary focus of this investigation centers on the utilization of real-time data processing techniques. Numerous prior studies have employed complex event processing to handle real-time events across various research domains (Baeth, 2018; Sudan, 2020; Aktas, 2020; Uzun Per, 2021; Dhaouadi, 2018; Can, 2023; Pinar, 2021;Cansiz, 2020). Our examination encompasses diverse distributed system architecture research areas, including service-oriented architectures (Tufek, 2018; Aktas, 2005; Dundar, 2021). However, this study uniquely delves into the methodologies for event-based distributed system architectures. Prior research has also explored the analysis of click-stream data to gain insights into user navigational behavior (Uygun, 2020), (Olmezogullari, 2020, 2022). In the course of this study, we developed a prototype software, deviating from the norm in the literature where other studies commonly assess the quality of such prototypes (Sahinoglu, 2015), (Kapdan, 2014). In contrast, our study does not explicitly address software quality as it falls beyond its scope.

Methodology

As part of the project, a customer actions business process proposal is being made to take actions on customer behavior according to certain rule sets by analyzing the customer’s actions in the application. The software methodology for the proposed process is shown in Figure-1. The process analyzes the data from customer actions. As a result of the analysis, many informative contents are presented to customers, such as the results of their actions, information about new campaigns, gains, and advantages. The activities carried out during the development of the proposed business process software can be described as follows:

• Project management activities: Tracking, reporting, and coordinating project processes from the beginning of the project to the transition to live environment.

• Analysis activities: Addressing the purpose, requirements, and potential risks of the project.

• Development activities: In the development activity, developing and testing software in accordance with the project requirements.

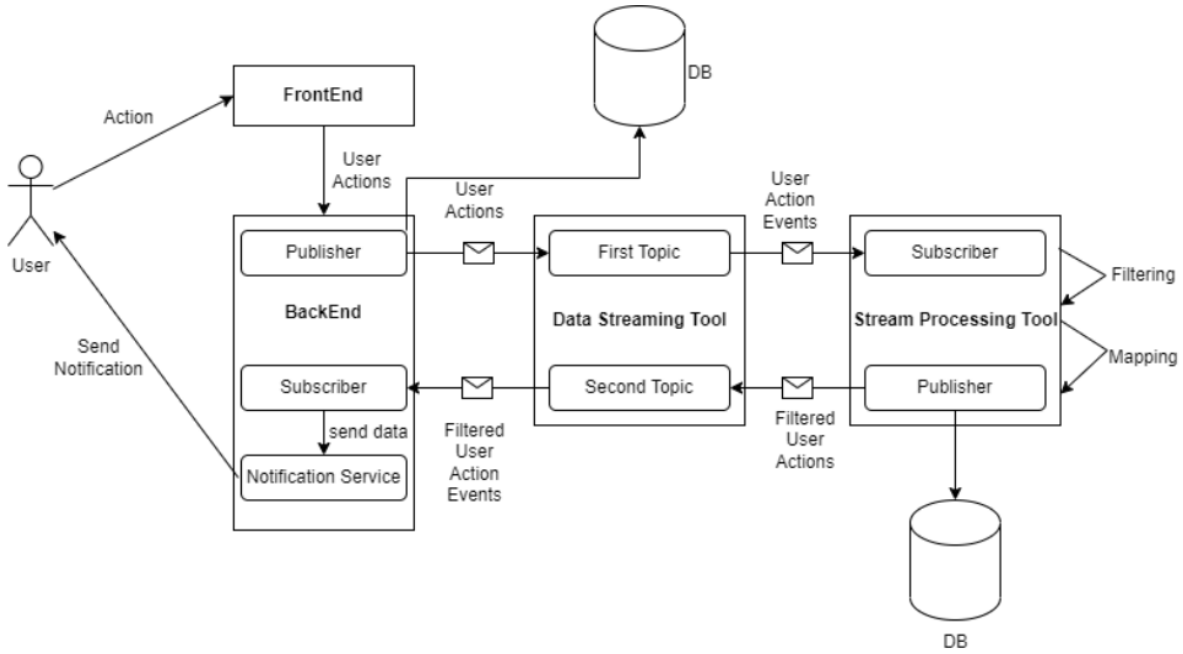

Figure 1. Business process flowchart for the proposed methodology

The requirements were determined for the modules, integration, structure, database modeling, and architectural design of the software to be developed. A literature review was conducted on real-time data transfer and real-time data processing, which are at the core of the project. As a result of the literature review, the data streaming tool (Kafka) and the real-time data processing tool (Flink) to be used in the project were compared with their counterparts and the technologies most suitable for the purpose of the project were selected. The dependencies with the other modules to be integrated with the project were reviewed and their suitability was tested.

In the software development process, the technologies detailed in Literature Review were first installed on local machines and integrated with other modules. Then, the data belonging to the users was sent to the first topic of the Data Streaming Tool. By reading the data in the first topic from the Real-Time Data Processing tool, new data was produced by filtering the user events that meet certain conditions. The produced data was first converted and then sent back to the second topic of the Data Streaming Tool. The necessary actions were taken by reading this data from another module. All data entering the Data Streaming Tool is stored in an Unassociated Database for reporting. After the development on the local machine was completed, the project was taken to the test environment for testing. Since user actions are already fed by the front end, no front-end development was done. Only backend development was done. While developing the backend, an object-oriented programming language was used

The process of transitioning the project to live environment is ongoing. Tests are being conducted with comprehensive test scenarios. The findings obtained as a result of the test will be reported and corrections will be made. When the tests are successfully completed, live environment transitions will be made. An educational content will be created to explain 5 the project. This educational content will include technical information (coding, architectural approach, etc.). When the transition to live environment is completed, the documents will be updated

Prototype Application and Evaluation of Application

A prototype project has been developed in accordance with the business process architecture given in the methodology section. The project will test the speed of real-time data processing, the accuracy of data, and the success of filtering.

In this context, Apache Kafka is used for real-time event transfer and Apache Flink is used for real-time data processing. The project is written in Java. There are libraries for Kafka and Flink in Java. User actions transferred to a different module via FrontEnd are converted to JSON string and written to Kafka topic via a producer. The data recorded in Kafka topic is also written to DB for reporting purposes. A distributed structure was created by setting up a separate server for Apache Flink. Thanks to this structure, Flink Server became independent. The data sent from a different module is read and converted to a Java object by a subscriber. Then, the data was passed through appropriate filters, the data that passed through the filter was grouped, and it was converted back to JSON String and written to a different topic by a subscriber. The data written to the different topic was also recorded to DB for reporting purposes. The filtered events from the module that sends the user movement are read by a subscriber and notification is sent to the user by the Notification service.

By increasing the number of user movements, the time it takes for the project to respond will be observed. In the first stage, the performance and status of the system will be observed by simulating the flows in the live environment. Then, it will be observed how the increase in data amount will affect the system, and all these observations will be reported.

The software to be produced within the scope of the project will be evaluated according to the following success criteria:

• Real-time data processing speed: The speed at which the project can process data coming in real time will be evaluated. This criterion will be important for determining the performance and scalability of the project.

• Data accuracy: The accuracy of the data processed in the project will be evaluated. Sending an inappropriate notification to the customer can negatively affect the customer experience. Additionally, if the notification is not sent to the customer, the customer may not be aware of the gains, actions, and benefits.

• Filtering success: It is related to data accuracy. The ability of the project to apply appropriate filters to the data will be evaluated.

• Customer satisfaction: The extent to which the project increases customer satisfaction will be evaluated.

• Sales and revenue growth: Whether it increases demand for products will be evaluated. Since campaigns, benefits, and offers will be offered to users in parallel with 6 their actions, more effective marketing and sales strategies will be implemented. In such a case, it is thought that the number of customers and demand will increase.

Conclusion

In this paper, a business process proposal is made that processes customer actions in real time and provides notifications to customers. To demonstrate the usefulness of the proposed business process, a prototype application has been developed. The details of the application have been provided. The tests of the prototype application are ongoing. The main goal of the tests is to evaluate the performance of the application. After the tests are completed, a usable product will be created and its transfer to the live environment is aimed to increase customer satisfaction and customer experience.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to authors.

Acknowledgements

* This article was presented as an oral presentation at the International Conference on Technology (www.icontechno.net) held in Antalya/Turkey on November 16-19, 2023.

* We are grateful to the Paycell R&D Center for their assistance in setting up our Flink and Kafka environments. We also thank Prof. Dr. Mehmet Sıddık Aktas for his contributions and guidance to this study.

References

Aktas, D. E., & Aktas, M. S. (2020). Real-time pattern detection methodology for monitoring student behaviour on e-learning platform in the field of financial sciences: Case study. *28th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Aktas, M., Aydin, G., Donnellan, A., Fox, G., Granat, R., Lyzenga, G., McLeod, D., Pallickara, S., Parker, J., Pierce, M., Rundle, J., & Sayar, A. (2005). Implementing geographical information system grid services to support computational geophysics in a service-oriented environment. *NASA Earth-Sun System Technology Conference.* University of Maryland, Adelphi, Maryland.

Apache Flink. (2023). Apache Flink documentation. Retrieved from https://flink.apache.org/

Apache Kafka. (2023). Apache Kafka documentation. Retrieved from https://kafka.apache.org/

Baeth, M. J., & Aktas, M. S. (2017). Detecting misinformation in social networks using provenance data. *13th International Conference on Semantics, Knowledge and Grids*.

Baeth, M. J., & Aktas, M. S. (2018) An approach to custom privacy policy violation detection problems using big social provenance data. *Concurrency and Computation: Practice and Experience 30*(21).

Can, A. B., Zaval, M., Uzun-Per, M., & Aktas, M. S. (2023). On the big data processing algorithms for finding frequent sequences. *Concurrency and Computation: Practice and Experience*, 1-17.

Cansiz, S., Sudan, B., Ogretici, E., & Aktas, M. (2020). Learning from student browsing data on e-learning platforms: Case study, computer science and information systems, *ACSIS, 20*, 37–44.

Carbone, P., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 36*(4).

Dhaouadi, J., & Aktas, M. (2018). On the data stream processing frameworks: A case study. *3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 104-109). IEEE.

Dundar, B., Astekin, M., & Aktas, M. S. (2021). A big data processing framework for self-healing internet of things applications. *IEEE International Conference on Big Data (Big Data),* 2353-2361.

Kapdan, M., Aktas, M., & Yigit, M. (2014). On the structural code clone detection problem: A survey and software metric based approach. *Computational Science and Its Applications–ICCSA 2014: 14th International Conference*.

Olmezogullari, E., & Aktas, M. S. (2020). Representation of click-stream datasequences for learning user navigational behavior by using embeddings. *IEEE International Conference on Big Data (Big Data),* 3173-3179.

Olmezogullari, E., & Aktas, M. S. (2022). Pattern2Vec: Representation of click-stream data sequences for learning user navigational behavior. *Concurrency and Computation: Practice and Experience 34*(9).

Pinar, E., Gul, M. S., Aktas, M. S., & Aykurt, I. (2021). On the detecting anomalies within the clickstream data: case study for financial data analysis websites. *6th International Conference on Computer Science and Engineering (UBMK*) (pp. 314-319). IEEE.

Sahinoglu, M., Incki, K., Aktas, Mehmet S. A. (2015). Mobile application verification: A systematic mapping study. *Computational Science and Its Applications- ICCSA 2015: 15th International Conference.* Banff, AB, Canada,

Sudan, B., Cansiz, S., Ogretici, E., & Aktas, M. S. (2020). Prediction of success and complex event processing in e-learning. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE*) (pp. 1-6). IEEE.

Tufek, A., Gurbuz, A., Ekuklu, O. F., & Aktas, M. S. (2018). Provenance collection platform for the weather research and forecasting model. *14th International Conference on Semantics, Knowledge and Grids*.

Uygun, Y., Oguz, R. F., Olmezogullari, E., & Aktas, M. S., (2020). On the large-scale graph data processing for user interface testing in big data science projects. *IEEE International Conference on Big Data (Big Data)*, 2049-2056

Uzun Per, M., Can, A. B., Gurel, A. V., & Aktas, M. S. (2021). Big data testing framework for recommendation systems in e-science and e-commerce domains. *IEEE International Conference on Big Data (Big Data)* (pp. 2353-2361). IEEE.

Uzun Per, M., Gurel, A. V., Can, A. B., & Aktas, M. S. (2022). Scalable recommendation systems based on finding similar items and sequences. *Concurrency and Computation: Practice and Experience, 34*(20).

Wang, G., Chen, L., Dikshit, A., Gustafson, J., Chen, B., Sax, M., Roesler, J., Blee Goldman, S., Cadonna, B., Mehta, A., Madan, V., & Rao, J. (2021). Consistency and completeness: Rethinking distributed stream processing in Apache Kafka. Proceedings of the *2021 International Conference on Management of Data*.

Wiatr, R., Słota, R., & Kitowski, J. (2018). Optimising Kafka for stream processing in latency sensitive. *Procedia Computer Science, 136,* 99-108.

Yildiz, E. C., Aktas, D. E., Unal, E., Tuzun, H., & Aktas, M. S. (2020). Management of virtualization technologies with complex event processing. *International Conference on Electrical, Communication, and Computer Engineering* (ICECCE) (pp. 1-4). IEEE.

Author Information

Alper Bozkurt
Turkcell Odeme ve Elektronik Para Hizmetleri A.S.
(Paycell Research and Development Center)
Aydınevler Mah. Ismet Inonu Cad. Turkcell Blok No:36 34854 Maltepe/ Istanbul, Turkey
Contact e-mail: *alper.bozkurt@turkcell.com.tr*

Furkan Ekici
Atmosware Teknoloji Egitim ve Danısmanlık A. S.
Aydınevler Mahallesi Inonu Caddesi No:20 Kucukyalı Ofispark, D: B Blok, 34854 Maltepe, Istanbul, Turkey

Hatice Yetiskul
Turkcell Odeme ve Elektronik Para Hizmetleri A.S.
(Paycell Research and Development Center)
Aydınevler Mah. Ismet Inonu Cad. Turkcell Blok No:36 34854 Maltepe/ Istanbul, Turkey

To cite this article:

Bozkurt, A., Ekici, F. & Yetiskul, H. (2023). Utilizing Flink and Kafka technologies for real-time data processing: A case study. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 24*, 177-183.