Contextual Data Stream Processing Overview, Architecture, and Frameworks Survey

Mohamed Bennawy[*1], Passent El-Kafrawy [**2]

[*] *School of Information Technology and Computer Science,*
Center of Informatics Science, Nile University, Giza 12588, Egypt.
m.zohair@nu.edu.eg
pelkafrawy@nu.edu.eg

Abstract: ***Event stream processing (ESP) is a data processing methodology which tackle online processing for a variety of events. Recently stream processing witnessed a huge interest in both academic research and corporate use cases. As a consequence, for the extremely huge data sources recently generated and diversely used. Data sources vary from social media feeds, news articles, internal business transactions, IoT devices logs, ... etc. Academically, a lot of research papers discuss how to deal with enormous cloud of events with different data structures such as text, video, logs, transactions, … etc. Also, research is concerned with different streaming platforms technologies; and evaluates the weakness and strength points of each. Researchers studied aside how to best utilize the platform within different use cases. From corporate point of view, decision makers ask about how to best utilize those events with minimal delay in order to 1) uncover insights in real-time, 2) mine textual events, 3) recommend decisions. This requires a mix of machine learning, stream and batch processing technologies which are typically optimized independently. However, combining all technologies by building a scalable real-world application is a challenge. In this paper, we shall discuss state of the art event stream processing technologies by summarizing definition, data flow architectures, textual use cases, frameworks and architecture best practice. Furthermore, we would discuss how to combine event stream processing with textual events and sentiment analysis to enhance a recommendation model outcome.***

Keywords: ***Event Stream Processing, Recommendation System, Sentiment analysis, Textual, Text mining, Kafka, Streaming, Machine Learning, Data Science, Apache Spark***

1 INTRODUCTION

Handling a stream of *real time events* continuously, is a huge benefit for any company. Currently in most corporates, advanced analytics for business intelligence analysis is done in offline batch architecture or semi-batch architecture. This cause these advanced analytics to be delayed one day than actuals. Consequently, important insights are not on time when needed. Event stream processing helps in getting fresher insights and faster reactions. Thus, decision makers will be able to give on time feedback and have full updated knowledge about the business performance and downsides. Two recent articles forecast that Complex Event Processing (CEP) market will grow 28.74% compound annual growth rate (CAGR) from 2021 till 2026 [1][2] while streaming analytics market CAGR will grow 20.6% by 2026 [3][4].

First of all, the main component of our research is an ***event***, thus the definition of event objects (also known as "event messages", or "event tuples") are objects that are acquired as an event through a specific time window. The data included in such objects commonly include the type of the event, time of the activity, the event location, trigger of the event, and other metadata. Therefore, a stream consists of a series of event objects, in chronological order of arrival, **Figure 1** [6].

Figure 1: A stream of events in sequence from e_1 to e_n [6]

Events within any large company can be categorized into three main types. The first is business transactions, are business related transactions or logs in different platforms. Identified by Customer account describing different key performance indicators. Examples: customer orders, insurance claims, bank deposits or withdrawals, customer address changes, telecommunication calls, delivery details, airline reservations, or transaction statements [7]. The second is textual reports, external sources of data to the company that are essential for decision makers [7]. Examples: tweets, news articles, market data, weather reports, and social media posts, Facebook and Linkedin messages. The third, and fastest growing is Internet of Things (IoT) data, that is data gathered from sensors produced by Physical devices [7]. Examples: GPS-location data

from vehicles or cellular, temperature or weather data, RFID tag readings and patient monitors, motion and body temperature [7].

The purpose of performing event stream processing in any of the three types is to enable businesses to react and take decisions in real-time. The stream of events requires advanced analytics that cannot be accomplished with normal machine learning models or common enterprise support systems. Stream analytics is known as event stream processing or complex event processing.

2 DATA ARCHITECTURE

The history of continuous event streams is reviewed in this section providing the evolution of business data flow frameworks. The advancement of this type of Data processing mechanisms had been evolved in three phases:

A. The classic era— since 2003 by Ralph Kimball, data in warehouses handled in batch loads and by time became big data. In this era most of the business decision(s)/analytics were processed offline and delayed from operation, **Figure 2** [8].

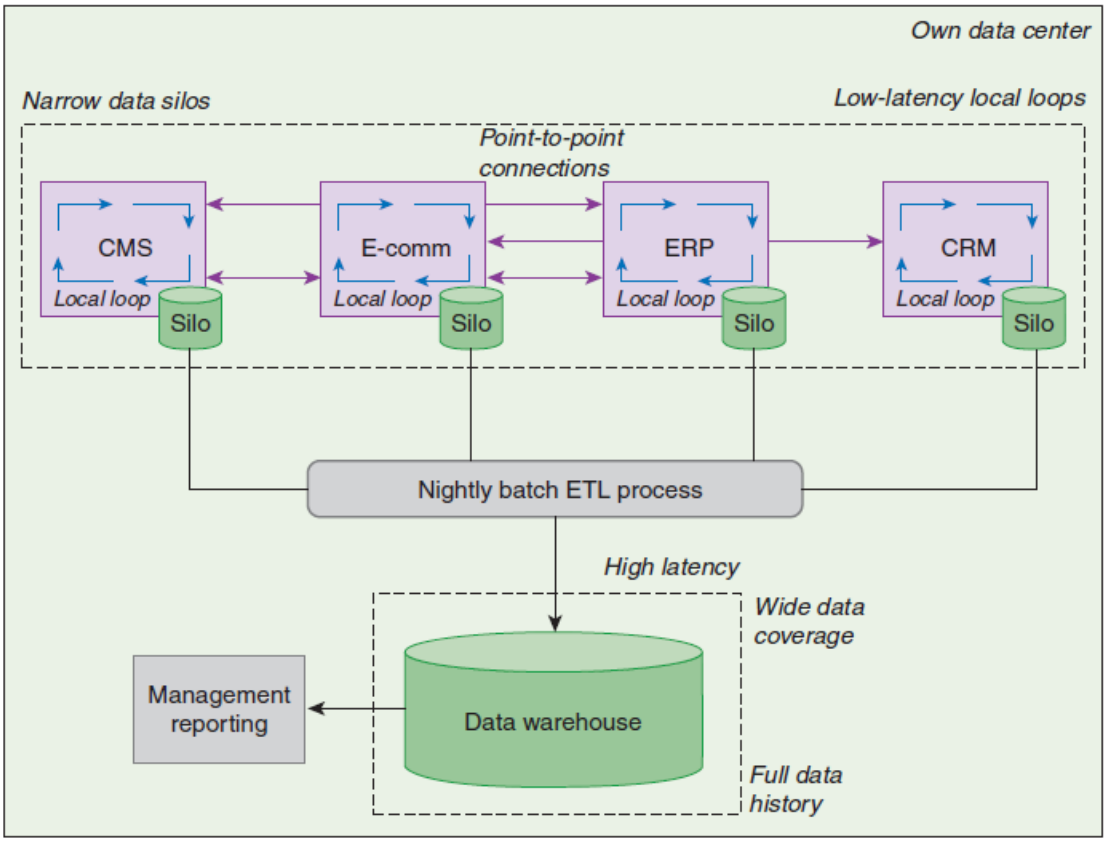

Figure 2: Classic era diagram [8]

Data were extracted in overnight batches through ETL (extract, transform and Load) processes extracting data from the data silos. Then the data are transformed or modelled for reporting, to be sent back to the data warehouse. The major drawback of the classical era is the high latency in reporting decisions. The batches were usually developed based on the star schema of fact and dimension data marts [8].

B. The hybrid era—current implementation of data flow is handled between batch loading and online streaming **Figure 3**.

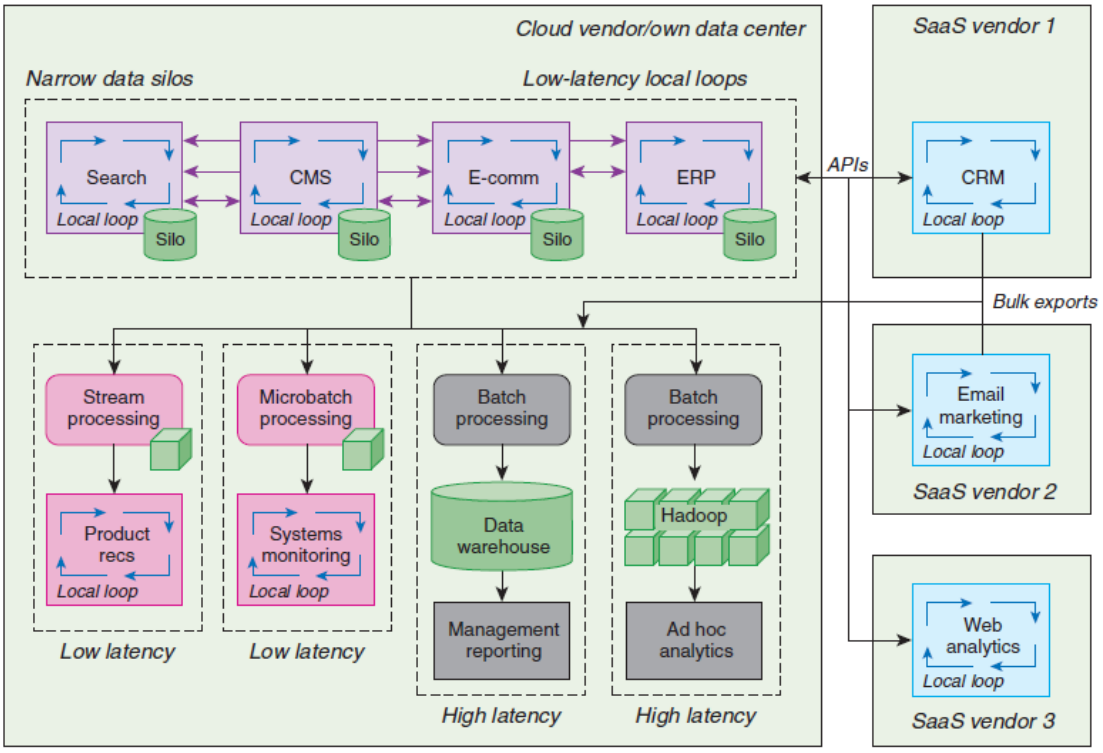

Figure 3: Hybrid era diagram [8]

In the hybrid era external dependencies were added; introduced Hadoop as a new high-latency and new low-latency data pipelines for use cases such as systems monitoring and product recommendations. Hybrid era pain points are no single point of truth, decision and solutions are fragmented and analysis either have low latency or wide data coverage, but not both [8].

C. *The unified era*—evolution in the architecture by emerging online streams in a unified log and enable continuous event processing for **Figure 4** [8].

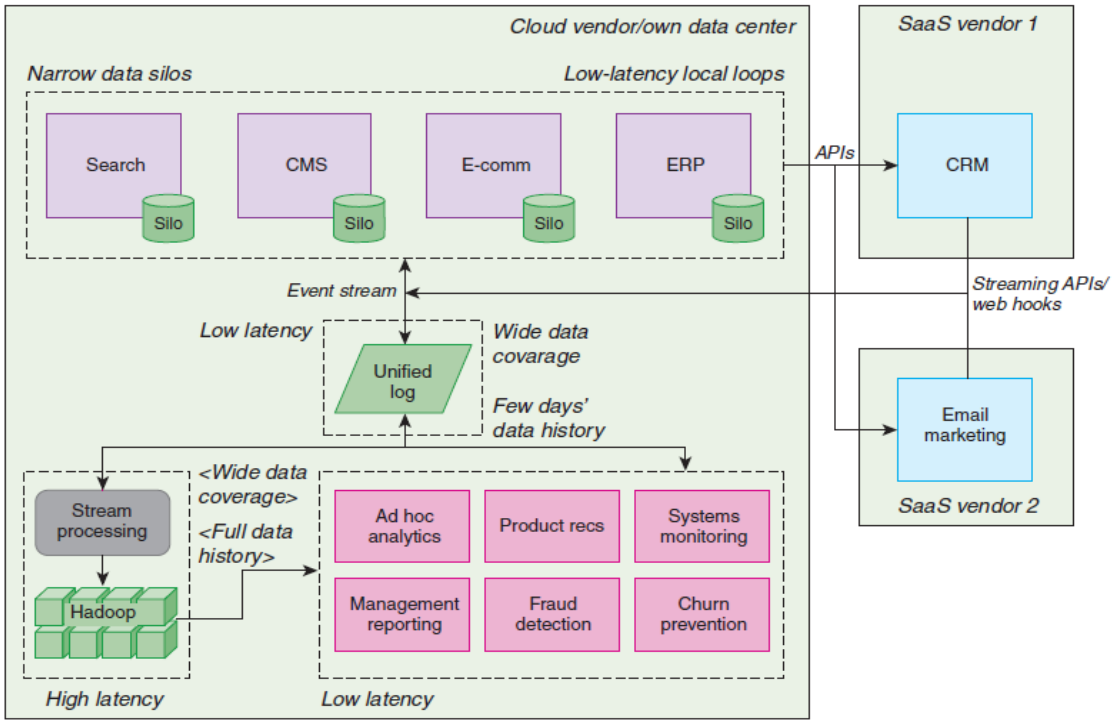

Figure 4: Unified Era Diagram [8]

The architecture emphasis on a unified log and a longer-term archive of events in Hadoop. The data architecture is now much simpler, with far fewer point-to-point connections, and all of our analytics and decision-making systems now working off a single version of the truth [8].

3 USE CASES

Making use of unified era architecture and the newly introduced streaming capabilities, a lot of use cases for event-driven applications will shine. Listed below list of use cases that will evolve with event stream processing capabilities [5]:

A. Manufactures
- Root Cause identification for any product issues by text analytics in logs across different sources [32].
- Market share analysis by understanding the competition products and shares [31].
- Customer social heartbeat, text mining for any social media feeds to understand the trending topics, identify influencers [35].

B. Government:
- Fraud emerging concerns that would help in shaping the policy.
- Public sentiments for unmet needs.
- Text Summarization [36].
- Smart cities [37].

C. Customer Care
- Conversational AI, Speech recognition to understand customer needs with on call virtual assistant with the ability to perform actions.
- Speech and Voice recognition, also known as speech to text (SST) to transcript to give a complete log with the call reasons and boost analytics capabilities.
- Proactive call deflections.

D. Finance
- Stock prices prediction, using NLP to summarize the web-based financial news aside with stock prices historical data.
- Credit scoring, NLP can assist in credit scoring by extracting relevant data from unstructured documents such as loan documentations, income, investments, expenses, etc. and feed it to credit scoring.

E. Healthcare
- Social media data to monitor and predict patients, streaming patient health attributes via social media and predicting the health status using machine learning [22-24] [27, 28].
- Find similar patterns in doctor reports [25, 26].
- Identify patterns in patients claims data for health insurance companies [29].

4 FRAMEWORKS

The software tools used in event stream processing can be categorized under three main subtypes as *event processing* platforms, *distributed stream computing* platforms and *complex event processing* (CEP) libraries/ Engines. The first category are event stream processing platforms where they provide high-level programming models such as high-level event processing languages (EPLs) and built-in functions for correlation, event filtering, and abstraction. As an example, Tibco stream base, IBM infosphere Stream [46], Data Torrent RTS [47], Amazon Kinesis [42], SQL Stream [33] and Fujitsu Interstage CEP [48]. The second category is Distributed Stream Computing Platforms/ Engines (DSCPs / DSPEs), where the main difference is providing explicit support for distribution across multiple nodes in a cluster. Example, Apache Storm [12], Spark Streaming [5], Apache Samza [19], Apex and Apache Flink [9]. Finally, the last category Complex event processing libraries / Engines which specially focus on the detection of complex patterns and events. As an example, for CEP libraries or engines Esper [34], Siddhi [38], ruleCore [39] and Cayuga [31].

According to **event stream processing** methodology in handling the stream of events, event stream processing frameworks implement any Streaming framework in two main categories:

A. Native Streaming: which reflect continuous processing for the events from different streams as soon as it arrived without any latency or waiting time window Figure 5. The continuous running processes are called operators or tasks or bolts based on the framework used. Examples: Storm [12], Flink [9], Kafka Streams [41], Samza [19].

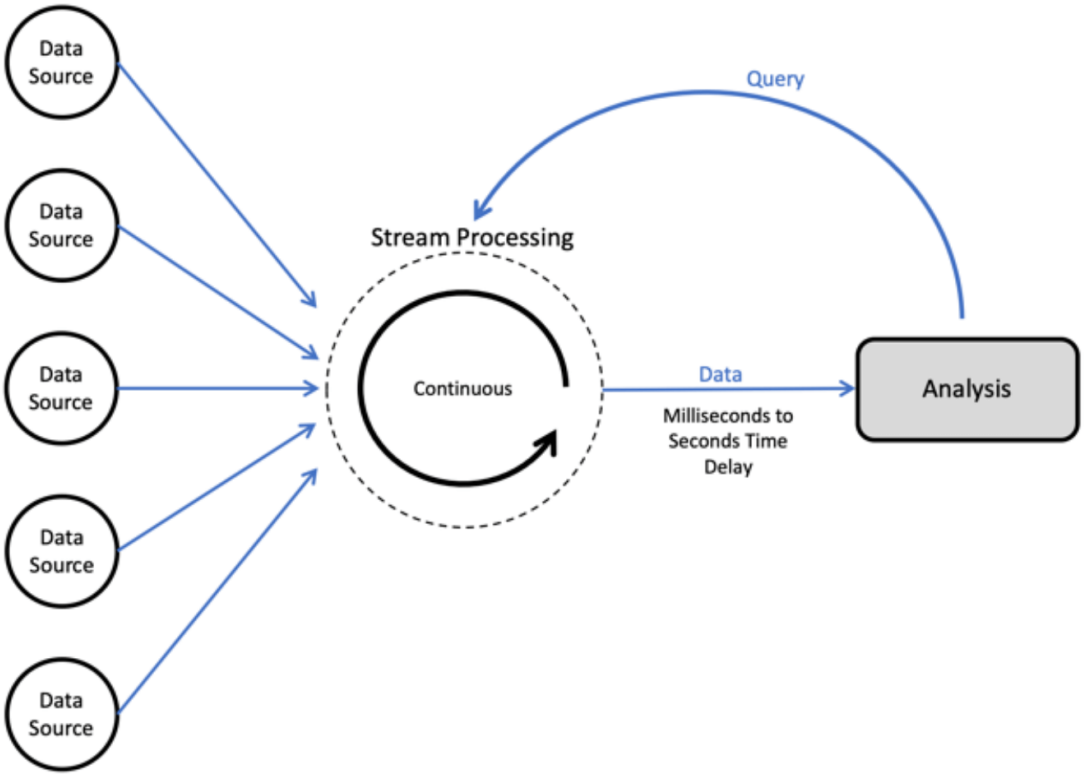

Figure 5: Native Streaming Diagram [11]

B. *Micro-batching*: rather than processing the events as soon as it arrives, the events are grouped in a predefined time window, and which called batched and then the processing is applied on the whole batch. It is known as fast batching as it is delayed few seconds than the event time **Figure 6** ex) Spark Streaming [5].

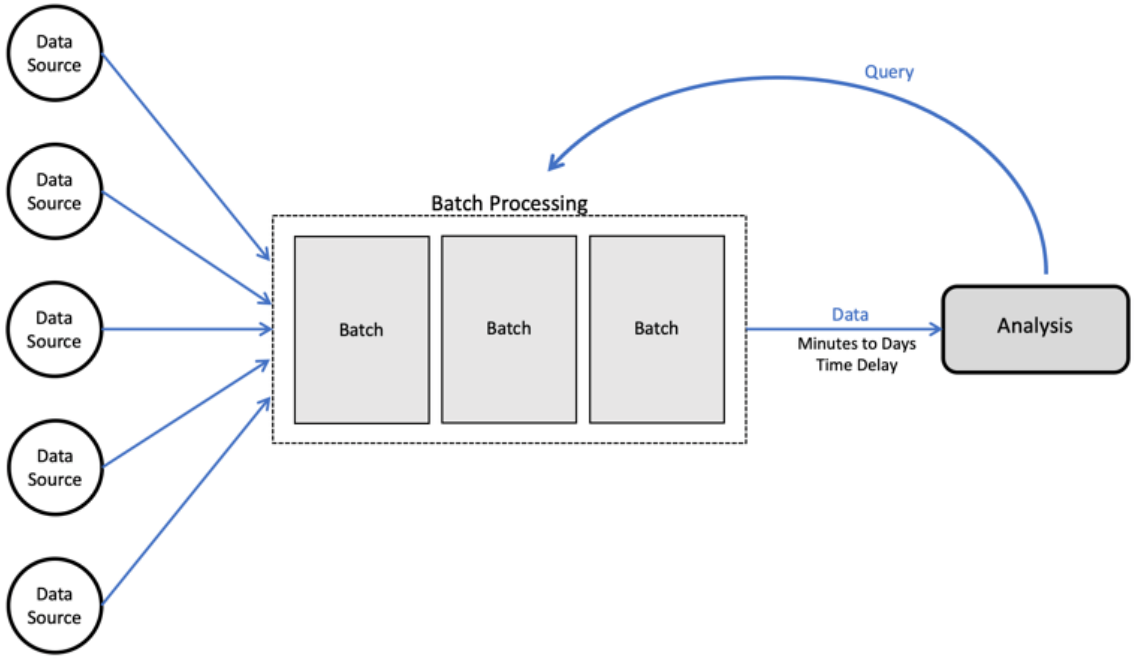

Figure 6: Micro Batching Streaming Diagram [11]

The main differences between the two methodologies, are mainly on fault tolerance, throughput, latency and state management. In *native streaming* the events are processed as soon as it arrives which means minimum latency, high throughput, easy state management, and hard to achieve fault tolerance. On the other hand, *micro batching* methodology is quite opposite because it groups the events into a small batch and then start processing. Therefore, micro batching has higher latency, lower throughput, harder to maintain state management, and better fault tolerance.

Based on the use case business, processing logic, latency constraints and deployment environment; a decision maker would choose the best methodology and framework to work with. Spark-Streaming [5], Apache Samza [19], Apache Flink [9], Trill, Apache Apex, etc. are examples for DSCPs which has both batch and stream processing capabilities. They have resulted from efforts made to develop unified processing frameworks for big and fast data. Most of these systems allow the users to specify an expected level of latency while maximizing the system's throughput [31].

5 ARCHITECTURE

Most streaming stacks are still built on an assembly line of open-source and proprietary solutions. Architecture is dynamic and mainly defined based on multiple aspects. use case scope, input data sources, data storage, data retention, integration points for the output and the running frequency are major aspects to define the perfect architecture. Answering these questions will help in shaping architecture and choosing technologies. First of all, identifying the data sources that will help in the feature selection process. Secondly, choosing convenient stream processing platform; to perform ETL (Extract, Transform and load) the data on time and make needed transformation if needed. Then, store the loaded data in the data lake storage with a predefined retention interval. Finally, applying machine learning logic to get the output which vary from prediction, semantic analysis, recommenders, ...etc. the output is then integrated with the real time applications to visualize the output or take necessary actions.

To wrap up, architecture has four main components 1) data sources 2) streaming processes and Storage 3) data science or machine learning logic 4) real time application integration. Figure 7 highlights the main components for a streaming use case with sample tools and logical workflow for any use case.

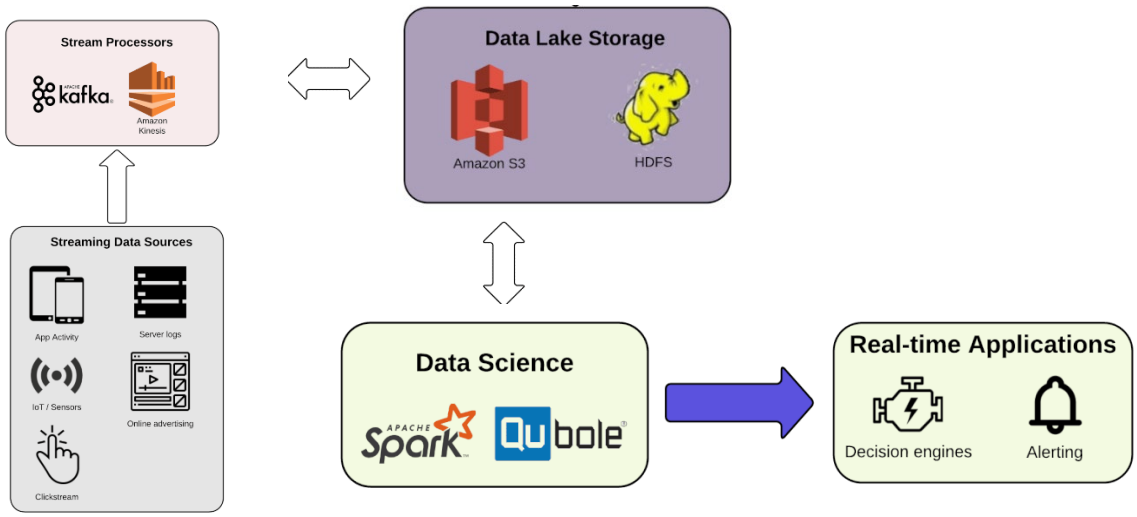

Figure 7: Main Components in event streaming architecture

The data sources vary based on use case and solution eco-system. One of the commonly used data sources is the clicks in a web page or application, because it reflects indirectly customer feedback and interest. Another example, social media feeds and tweets that reflect customer opinion in textual format. In addition to, Server logs, sensors readings from internet of things (IoT) devices. Data sources reflects customer interest and opinion in real time using different data formats, devices and locations.

Streaming platform is responsible of availing feeds or events from data sources to data lake storage. Platforms are different in workload, frequency and transformation logic capabilities. Thus, an open-source platform like Apache Kafka [41], or Solution like Amazon Kinesis [42] can be considered as an event streaming platform. Streaming platform define the window size, fetching mode (batch / real time) and if any transformation logic on data.

Before consuming event stream, data need to be saved first into a data lake, i.e., Hadoop distributed file system (HDFS) [45], rational database management system (RDBMS), or NoSQL database, i.e., MongoDB [43]. Data volume, integration points and constraints define the data lake storage for your solution.

Afterwards, data science component implements utilize the streaming data with data science or machine learning algorithm. Data science platforms have multiple options either open-source or product based. As an example, Apache Spark [20] and Qubole [44]. The perfect platform should fulfill use case scope, consume reasonable time, and integrate output with the real time application. The real time application can be an alerting system or visualization tools to help in decision making for ease of use.

A. Session Based Recommender System

One of the common use cases, real time recommendation engines on a streaming data. Recommendation is the art of understanding targeted customer preferences and interests to help in proposing next best action. Having the ability to stream events in real time and understand contextual content, would boost the recommendation engine accuracy and give the right item to the right customer on the right time. Recommendation engine on streaming data identified as session-based recommenders and Probabilistic latent semantic analysis [40] where we recommend in real time with sentiment analysis to understand content and help in getting next best action.
Session-based recommendation systems (SBRSs) have appeared as a new paradigm of recommendation systems (RSs). Different from other RSs such as content-based RSs and collaborative filtering-based RSs which usually model long-term yet static user preferences, SBRSs aim to capture short-term but dynamic user preferences to provide more timely and accurate recommendations sensitive to the evolution of their session contexts.

B. Benefits of modern stream processing architecture

Implementation of modern streaming architecture have multiple advantages. First of all, it shrinks the need for large data engineering solutions. Secondly, it boosts the performance, high availability and fault tolerance. Furthermore, newer platforms are cloud based and can be deployed very quickly with no upfront investment. Finally, it avails huge flexibility and support.

Although event stream processing (ESP) is widespread in various domains and many related studies have been conducted, there are many inconsistencies in the area of ESP caused by the diverse descriptions, settings, assumptions and application domains. There is no unified framework that well categorize them and there are no unified problem statements for ESP. No systematic categorization for all the representative and state-of-the-art approaches for ESP.

6 CONCLUSION

Event stream processing will be part of any data architecture as most corporates are moving from the classic era architecture to the unified era architecture. Corporates and decision makers aim to best utilize cloud of events and data available in order to take decisions and get insights in real time rather than offline batches. Therefore, we introduced top use cases and frameworks that would benefit from the real time decision(s) produced from event stream processing capabilities. Regarding frameworks selection, multiple factors including deployment, ecosystem, security… etc. would decide the perfect framework for application and a detailed comparison is shared to illustrate the main differences between the listed frameworks. We provided the investigation through an evaluation. From this evaluation we were able to recommend an architecture and stream processor platform that suffices corporate's needs. Additionally, this evaluation provides a general overview of the leading architectures and stream processors differences. This allows for other businesses in a similar situation to draw beneficial value from its generality. Finally, one of the very promising use cases is the usage of event stream processing in real time text recommendations which will be studied further in the future.

REFERENCES

[1] 2020. Complex Event Processing (CEP) Market - *Global forecast to 2026. RESEARCH AND MARKETS.*

[2] Mordor Intelligence Web Site: https://www.mordorintelligence.com/industry-reports/global-complex-event-processing-cep-market-industry , (accessed July 2021)

[3] 2015. Streaming Analytics Market by Verticals - *Worldwide Market Forecast & Analysis (2015 - 2020).*

[4] Mordor Intelligence Web Site: https://www.mordorintelligence.com/industry-reports/event-stream-processing-market , (accessed July 2021)

[5] Stream Processing with Apache Spark – *ISBN: 978-1-491-94424-0*

[6] Complex Events Web Site: https://complexevents.com/2020/06/15/whats-the-difference-between-esp-and-cep-2/ , (accessed Aug 2021)

[7] Complex Events Web Site: https://complexevents.com/when-do-you-need-an-event-stream-processing-platform/ ,(accessed Aug 2021)

[8] Event Streams in Action Real-time event systems with Kafka and Kinesis by Alexander Dean, Valentin Crettaz (z-lib.org) - ISBN: 9781617292347

[9] Stream Processing with Apache Flink - ISBN: 978-1-491-97429-2

[10] Complex Events Web Site: https://complexevents.com/2019/06/09/eight-trends-in-event-stream-processing-may-2019/ , (accessed Sep 2021).

[11] Up Solver Web Site: https://www.upsolver.com/blog/streaming-data-architecture-key-components, (accessed Aug 2021).

[12] Apache Storm Web Site: https://storm.apache.org/releases/current/Tutorial.html , (accessed Aug 2021).

[13] Wikipedia Web Site: https://en.wikipedia.org/wiki/Apache_Storm , (accessed Aug 2021).

[14] Apache Heron Web Site: https://heron.incubator.apache.org/ , (accessed Aug 2021).

[15] Twitter Blog Web Site: https://blog.twitter.com/engineering/en_us/a/2015/flying-faster-with-twitter-heron, (accessed Aug 2021).

[16] Research Gate Web Site: https://www.researchgate.net/publication/319680334_Aging-related_Performance_Anomalies_in_the_Apache_Storm_Stream_Processing_System, (accessed Oct 2021).

[17] Engineering LinkedIn Web Site: https://engineering.linkedin.com/blog/2018/11/samza-1-0--stream-processing-at-massive-scale, (accessed Aug 2021).

[18] Martin Kleppmann Web Site: https://martin.kleppmann.com/papers/samza-encyclopedia.pdf, (accessed Sep 2021).

[19] Apache Samza Web Site: https://samza.incubator.apache.org/learn/documentation/latest/introduction/architecture.html, (accessed Sep 2021).

[20] Apache Spark Web Site: https://spark.apache.org/releases/spark-release-3-1-1.html , (accessed Aug 2021)

[21] Apache Nifi Web Site: https://nifi.apache.org/, (accessed Sep 2021).

[22] Fan Zhang, Junwei Cao, Samee U. Khan, Keqin Li, Kai Hwang, A task-level adaptive MapReduce framework for real-time streaming data in healthcare applications, *Future Generation Computer Systems*, Volumes 43–44,2015.

[23] Lekha R. Nair, Sujala D. Shetty, Siddhanth D. Shetty, "applying spark-based machine learning model on streaming big data for health status prediction", *Computers & Electrical Engineering*, Volume 65, 2018.

[24] De Silva D, Ranasinghe W, Bandaragoda T, Adikari A, Mills N, Iddamalgoda L, et al. "Machine learning to support social media empowered patients in cancer care and cancer treatment decisions." *PLoS ONE* 13(10): e0205855, 2018.

[25] Gabarron, Elia & Zubiete, Enrique & Rivera, Octavio & Wynn, Rolf. Diabetes on Twitter: A Sentiment Analysis. *Journal of Diabetes Science and Technology*. 13. 193229681881167. 10.1177/1932296818811679. 2018.

[26] Svenstrup, Dan & Jørgensen, Henrik & Winther, Ole. Rare disease diagnsis: A review of web search, social media and large-scale data-mining approaches". *Rare disease*.Vol. 3.e1083145. 2015.

[27] Plachouras, Vassilis & Leidner, Jochen L. & Garrow, Andrew. Quantifying Self-Reported Adverse Drug Events on Twitter: *Signal and Topic Analysis*. 1-10. 10.1145. 2016.

[28] Clark, Eric & James, Ted & Jones, Christopher & Alapati, Amulya & Ukandu, Promise & Danforth, Christopher & Dodds, Peter. A Sentiment Analysis of Breast Cancer Treatment Experiences and Healthcare Perceptions Across Twitter. 2018.

[29] Hager Ahmed, Eman M.G. Younis, Abdeltawab Hendawi, Abdelmgeid A. Ali, Heart disease identification from patients' social posts, machine learning solution on Spark, *Future Generation Computer Systems*, Volume 111, 2020.

[30] A. Hassan and A. Mahmood, Deep learning approach for sentiment analysis of short texts, in: *2017 3rdInternational Conference on control, automation and robotics (ICCAR)*, pp. 705–710, et al., April, 2017.

[31] Dayarathna, Miyuru & Perera, Srinath. (2018). Recent Advancements in Event Processing. *ACM Computing Surveys*. 51. 1-36. 10.1145/3170432.
[32] Na Mao and Jie Tan. 2015. Complex Event Processing on uncertain data streams in product manufacturing process. In *Advanced Mechatronic Systems (ICAMechS), 2015 International Conference* on. 583–588. RESEARCH AND MARKETS.
[33] SQL Stream Web Site: https://sqlstream.com/ , (accessed Nov 2021).
[34] Esper Tech Web Site: https://www.espertech.com/esper/ , (accessed Nov 2021).
[35] James Benhardus and Jugal Kalita. 2013. Streaming Trend Detection in Twitter. *Int. J. Web Based Communities* 9, 1 (Jan. 2013), 122–139.
[36] Xenofontas Dimitropoulos, Marc Stoecklin, Paul Hurley, and Andreas Kind. 2008. The Eternal Sunshine of the Sketch Data Structure. *Comput. Netw*. 52, 17 (Dec. 2008), 3248–3257.
[37] Muhammad Intizar Ali, Feng Gao, and Alessandra Mileo. 2015a. CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II. 374–389
[38] Siddhi Web Site: https://siddhi.io/ , (accessed Nov 2021).
[39] Fresh Meat Web Site: http://freshmeat.sourceforge.net/projects/rulecore , (accessed Nov 2021).
[40] Z. Hyung and K. Lee, "Recommending Music Based on Probabilistic Latent Semantic Analysis on Korean Radio Episodes," 2013 *Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 472-476, doi: 10.1109/IIH-MSP.2013.123.
[41] Apache Kafka Web Site: https://kafka.apache.org/ , (accessed Aug 2021).
[42] Aws Kinesis Web Site: https://aws.amazon.com/kinesis/ , (accessed Nov 2021).
[43] MonoDB Web Site: https://www.mongodb.com/ , (accessed Nov 2021).
[44] QUBOLE Web Site: https://www.qubole.com/ , (accessed Nov 2021).
[45] Apache Hadoop Web Site: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html , (accessed Nov 2021).
[46] IBM Web Site: https://www.ibm.com/docs/en/streams/4.1.1?topic=welcome-introduction-infosphere-streams , (accessed Nov 2021).
[47] Data Torrent Web Site: https://dt-docs.readthedocs.io/en/stable/#:~:text=DataTorrent%20RTS%2C%20built%20on%20Apache,data%20ingestion%20and%20distribution%20features. , (accessed Nov 2021).
[48] Fujutsu Web Site: https://www.fujitsu.com/jp/products/software/middleware/business-middleware/interstage/products/bigdataceps/function/ , (accessed Aug 2021).
[49] Datanami Web Site: https://www.datanami.com/2019/05/30/understanding-your-options-for-stream-processing-frameworks/ , (accessed Nov 2021).
[50] Making Sense of Stream Processing - The Philosophy Behind Apache Kafka and Scalable Stream Data Platforms, Martin Kleppmann - ISBN 978-1-491-93728-0
[51] Up Solver Web Site: https://www.upsolver.com/blog/popular-stream-processing-frameworks-compared , (accessed Nov 2021)
[52] Apache Flink Web Site: https://flink.apache.org/news/2016/04/06/cep-monitoring.html , (accessed Nov 2021)

BIOGRAPHY

Mohamed Bennawy received the bachelor's degree in computer science and information technology from Helwan University, in 2007. He has been working as software developer for 2 years from 2007 till 2009. Then join telecommunication corporate for 8 years in which 5 years as a business intelligence supervisors and 3 years as a business intelligence competency center senior supervisor. Later in 2017, He joined banking industry in data analytics team as principal solution architect for 1 year. Finally, he joined back telecommunication corporate as a data science and analytics senior manager. He had professional diploma in big data and machine learning from Nile University in 2016. His master's in informatics is in progress since 2017. His research interest in analytics, business intelligence, data science, machine learning, recommendation systems and visualization.

Passent M. El-Kafrawy received the bachelor's degree from the Computer Science and Engineering Department, American University, Cairo, the master's degree from the Faculty of Science, Menoufia University, and the Ph.D. degree in Computer Science and Engineering, in the field of computational geometry and artificial intelligence, from the University of Connecticut, USA, in 2006. She joined the Information Technology and Computer Science School, Nile University, in 2019. She has been a Professor, since 2018. Then she taught at Eastern State University of Connecticut for one year. In 2007, she has worked as an Assistant Professor with the Mathematics and Computer Science Department, Faculty of Science, Menoufia University. In 2011, she joined the Computer Science and Engineering Department, American University, as an Adjunct Professor. She has appointed as an Associate Professor in 2013. She has over 45 publications and editor in three books. She has been supervising several research studies between Ph.D. and M.Sc. in the field of natural language processing, semantic knowledge, bioinformatics, big data analytics, and knowledge mining and acquisition. She is a member of the Egyptian Society of Language Engineering and the Editor in Chief of the Journal of Egyptian Language Engineering. She has joined the IBRO School for neurodegenerative physician training organized by ENND and personalized medicine workshop organized by AUC. Her research interests include software engineering, bioinformatics, big data analytics, machine learning, and cloud computing.

ARABIC ABSTRACT

عرض لتقنيات معالجة وتحليل البيانات المتدفقة فى الحال

محمد بناوى،[1*] بسنت الكفراوى[2**]

* *كلية تكنولوجيا المعلومات وعلوم الكمبيوتر ،*

** *مركز علوم المعلوماتية، جامعة النيل، الجيزة* ,12588 *القاهرة .*

[1]m.zohair@nu.edu.eg
[2]pelkafrawy@nu.edu.eg

ملخص:

تشهد معالجة تدفق البيانات مؤخرًا اهتمامًا كبيرًا بالبحوث الأكاديمية والشركات. حيث تراقب الشركات سحابة من البيانات تختلف من مواقع التواصل الاجتماعي، والمقالات الإخبارية، والمعاملات التجارية الداخلية، وسجلات أجهزة إنترنت الأشياء، ... إلخ. يتطلب لأصحاب القرار في الأعمال الاستخدام الأفضل لتلك البيانات مع أدنى حد من التأخير من أجل 1) الكشف عن المؤشرات سريعا، 2) التنقيب عن الأحداث النصية، 3) التوصية بالقرارات.

تتيح التطورات الحديثة في التعلم الآلي معالجة البيانات وتحليل البيانات بسرعة فائقة. من الناحية الأكاديمية، ناقشت الكثير من الأوراق البحثية كيفية التعامل مع السحابة الهائلة للبيانات ببيانات مختلفة مثل النصوص والفيديو والسجلات والمعاملات وما إلى ذلك. أيضًا. من وجهة نظر الشركة، يسألون عن أفضل طريقة للاستفادة من هذه الأحداث بأقل قدر من التأخير.

يتطلب هذا مزيجًا من تقنيات التعلم الآلي والمعالجة التي يتم تحسينها عادةً بشكل مستقل. ومع ذلك، فإن الجمع بين جميع التقنيات وبناء تطبيق واقعي قابل للتطوير يمثل تحديًا. في هذه الورقة، سنناقش أحدث تقنيات معالجة تدفق البيانات من خلال تلخيص التعريف، وبنى تدفق البيانات، وحالات الاستخدام النصية.

الكلمات الدالة :*معالجة تدفق الأحداث، نظام التوصيات، تحليل المشاعر ، النص ، التنقيب عن النص، أباتشي كافكا ، التعلم الآلي ، علوم البيانات ، أباتشي سبارك*