International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Event-Driven Data Processing for Business Intelligence Reporting

Girish Ganachari

Email: girish.gie[at]gmail.com

Abstract: ***When comparing event-driven data processing versus batch processing, the former adds important insights. Routers, consumers, and producers are three crucial elements of event-driven architecture to collaborate to guarantee precise and prompt data analysis and transmission. Businesses may use real-time data analytics to find new patterns and trends. Moreover, the difficulties with complexity and inconsistent information. Businesses may handle efficient BI reporting and event-driven data processing by implementing strategies like separate data volumes and the auto-scaling mechanism.***

Keywords: Auto-scaling, batch processing, real-time data analytics, microservice architecture, functional hierarchy.

1. Introduction

In recent times, event-driven data processing in business management is considered a powerful approach towards business intelligence (BI) reporting as leverages the data processing focused on events compared to batch data processing based on scheduled intervals. This type of data processing can be useful in gaining more time-focused insights and enhanced decision-making procedures in business administration [1]. The key concepts such as the event-driven architecture, real-time data processing, and data pipeline designs are involved in the event-driven processing of data [2]. These set up key use cases in dynamic business decisions, pricing determination, real-time monitoring of sales, and detection of frauds effectively.

2. Literature Review

2.1 Overview of Event-driven data processing

A software development pattern referred to as event-driven processing of data, or event-driven design, enables systems to recognise, handle, control, and respond to events in real time as they occur. When something takes place, data about it is transmitted to all the applications, structures, and users who require it to respond instantly [3].

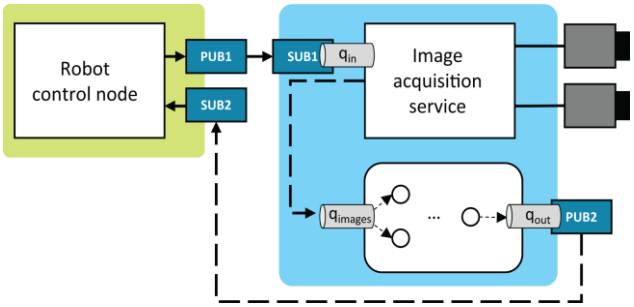

Figure 1.1: A framework for constructing event-driven processing of data
(Source: Semeniuta and Falkman, 2019)

Three essential elements comprise event-driven structures that are producers of events, routers of events, along with consumers of events [4]. The router receives an event published by a producer, analyses it, and then delivers it to users. Because both businesses and their services are not linked, they can be expanded, updated, or utilised separately. Therefore, it can be determined that event-driven designs involve real-time responsiveness that can be helpful in the transmission of the event data of the consumer of events.

2.2 Components of Event-driven Architecture

Significant events have been determined and event messaging is produced via an architecture driven by events. After being analysed by a message process engine, the corresponding messages are distributed to all the appropriate parties, who may evaluate the occurrences and decide whether or not to initiate action [8]. In addition, the occurrence processing system has the flexibility to initiate processes, call offerings, or store event information [10]. The functional hierarchy of the event-driven data architecture involves the levels or scales which include company management, business planning, and logistics management which can have discrete control, batch control, and continuous control that can be involved in the BI reporting [5] (***Refer to Figure 2.2***).

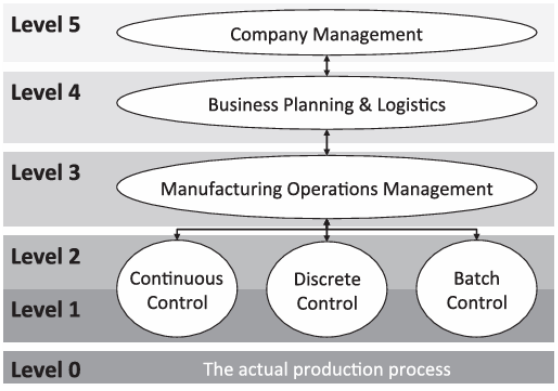

Figure 2.2: The functional architecture of event-driven data processing
(Source: Theorin *et al.*, 2016)

2.3 Technological advancements in event-driven data processing

The history of event-driven data processing begins with the Active Database period. Numerous event-driven data processing programs have emerged since subsequently. The

processing of streams as well as Complex Event Processing (CEP) constitute the two primary subfields of event-driven data processing [6].

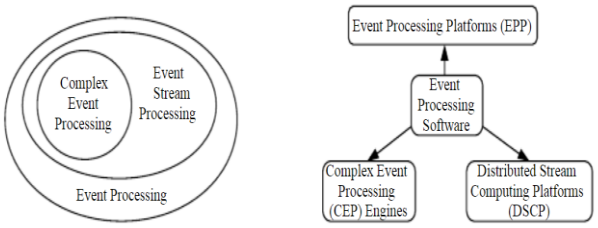

Figure 2.3: Technological advancements in event-driven processing of data
(Source: Dayarathna and Perera, 2018)

The first section, known as streaming or incident processing, is capable of supporting many forms of ongoing analytics, including joining, filtering, aggregating, enhancement, and classification. Similarly, event-driven data processing detects and reports compound events by utilising trends over basic event sequences [7]. Event-driven data processing has been used to serve as an operator in a general-purpose streaming system for processing in business management reporting [11]. On the contrary, it can be determined that general-purpose processing of events is now supported by event-driven data processing algorithms.

2.4 Challenges of event-driven data processing

One of the major challenges that can be determined in event-driven data processing is complexity. Intricacy can be defined in business administration intelligence reporting, maintaining synchronisation, controlling race situations, and resolving deadlocks, asynchronous event-driven systems can be complicated and necessitate exact programming [12]. Another challenge can be determined as the lack of data consistency [13]. Maintaining data consistency throughout multiple parts that attend to events may be challenging, particularly when those parts analyse the events at various times as well as places [14]. Therefore, it can be evaluated that these challenges can hamper the evaluation of the BI reporting by the companies.

3. Methodology

In order to conduct the results, the secondary qualitative data collection process has been applied. The conduct of the outcomes that are required to be involved in the study discussion, has been connected to the analysis of the event-driven processing of data that can be connected to the BI reporting focused on the prior articles. The secondary qualitative analysis helps the individuals to gain extensive insights based on the previous papers which can help to analyse similar contexts [15]. The articles which have been published in the timeframe of 2014-2019 are focused on peer-reviewed journals.

4. Results and Discussion

4.1 Impact of event-driven data process in BI reporting

Developing flexible architectures that are service-oriented and microservice-based platforms has made architectures based on events much more standard. Recognising the fundamentals of event-driven design is an advantageous ability that BI reporting may possess in the business toolbox. It might be beneficial to choose an architecture driven by events [16]. A loosely linked system that allows each component to expand, scale, as well as fail on its own is designed and constructed. When anything happens, a lot of different components and structures need to be elaborated. Even if different groups own systems, BI management needs to duplicate data between them [17]. One of the major implications that can be implied by the event-driven data process in BI reporting is the real-time insights gaining and the timeliness of access. Event-driven processing data compared to traditional data processing helps the business to handle the data after it has been generated and allows the companies to react to the determined changes along with trends in an instant mode [19].

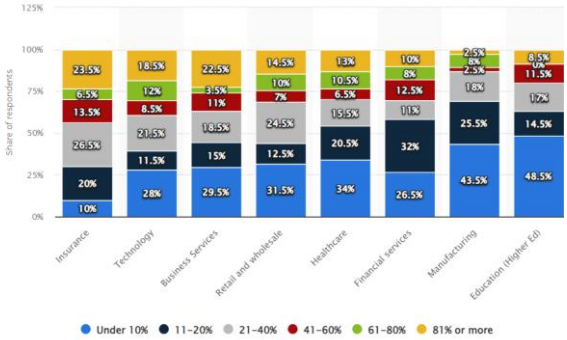

Figure 4.1: Implication of Business Intelligence (BI) in industries
(Source: Sherif, 2018)

In recent times, in many industries, BI reporting has been utilised, such as technology, insurance, healthcare, financial sectors, manufacturing, and others [18]. Therefore, it can be determined that to successfully drive the BI reporting, the event-driven data processing can be helpful in addressing the data collection that is processed in scheduled intervals

4.2 Advantages of event-driven data processing

One of the major advantages of implying event-driven data processing is the inclined flexibility as well as scalability. The architectures driven by the events are scalable inherently, and it allow to operation of the business intelligence reporting independently by the consumers, routers, and producers [20]. The system which utilises the BI reporting can effectively scale by addressing more components effectively as needed in the BI reporting [21]. This involvement of decoupling can also incline the possibilities of flexibility, and businesses can also update and expand their BI management without majorly disrupting the overall system [22].

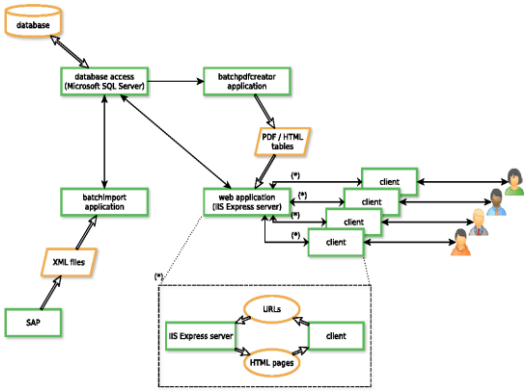

Figure 4.2: Flexibility and scalability of event-driven data processing
(Source: Semantic Scholar, 2016)

Moreover, improved decision-making is also one of the key advantages of the involvement of the event-driven data processing system that can be involved in the BI reporting system. Based on the provision of accurate as well as timely insights, event-driven architecture can help to enhance the process of decision-making [23]. Businesses can leverage the analytics of real-time data for identification of emerging patterns, and trends, and it also can detect anomalies so that the business can make proactive strategic adjustments

4.3 Strategies to imply event-driven data processing by companies

Choosing appropriate tools, as well as a proper technological track, can support the implementation of the event-driven architecture, and the real-time processing of data. The companies use stream-based processing frameworks such as Apache Flink, Apache Storm, or Apache Spark Streaming to process the streaming of data through event-driven processing to effectively implement BI reporting [24].

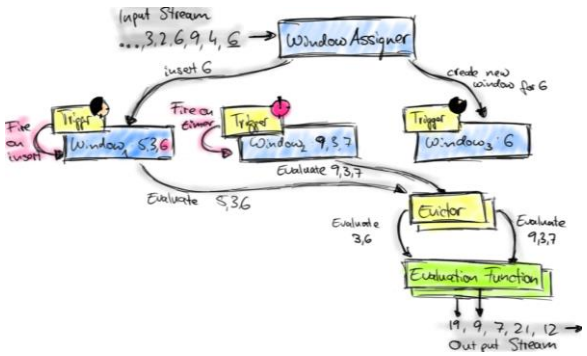

Figure 4.3: Apache Flink-based streaming frameworks
(Source: Apache Flink, 2015)

Another strategy that is implemented by the companies to be involved in BI reporting is the design of flexible and scalable architectures. The creation of the decoupled architectures allows the updating and scaling of the components connected to the event-driven data processing to handle different data volumes as well as velocities. Microservice architecture is preferred by companies to ensure that all the producers, consumers, and routers are scaled independently [25]. The implication of the auto-scale mechanism is connected to the microservice architecture, which can implement the validation of robust data. This can help in the error-handling process, and checking of consistencies in the management of real-time insights. Therefore, the strategies of microservice architecture, and handling of independent data volumes can be beneficial for the companies to successfully develop BI reporting through the implementation of event-driven data processing.

5. Conclusion

Event-driven data processing helps in adding significant insights compared to the traditional method of batch processing. The essential components of event-driven architecture such as routers, consumers, and producers work combined to ensure accurate and timely data analysis and transmission process. Real-time data analytics may be utilised by businesses to detect new trends and patterns. Furthermore, the challenges of intricacy and the lack of information consistency. Strategies such as the auto scale mechanism, and independent data volumes are useful for companies to implement event-driven data processing as well as handling effective BI reporting.

References

[1] Z. Milosevic, W. Chen, A. Berry, and F. A. Rabhi, “An open architecture for event-based analytics,” *International Journal of Data Science and Analytics*, vol. 2, no. 1–2, pp. 13–27, Oct. 2016, doi: https://doi.org/10.1007/s41060-016-0029-7.

[2] G. Ortiz, J. A. Caravaca, A. G. Prado, F. C. de la O, and J. B. Puig, “Real-Time Context-Aware Microservice Architecture for Predictive Analytics and Smart Decision-Making,” *IEEE Access*, vol. 7, pp. 183177–183194, 2019, doi: https://doi.org/10.1109/access.2019.2960516.

[3] O. Semeniuta and P. Falkman, “EPypes: a framework for building event-driven data processing pipelines,” *PeerJ Computer Science*, vol. 5, p. e176, Feb. 2019, doi: https://doi.org/10.7717/peerj-cs.176.

[4] S. Zhelev and A. Rozeva, “Using microservices and event driven architecture for big data stream processing,” *PROCEEDINGS OF THE 45TH INTERNATIONAL CONFERENCE ON APPLICATION OF MATHEMATICS IN ENGINEERING AND ECONOMICS (AMEE’19)*, 2019, doi: https://doi.org/10.1063/1.5133587.

[5] A. Theorin *et al.*, “An event-driven manufacturing information system architecture for Industry 4.0,” *International Journal of Production Research*, vol. 55, no. 5, pp. 1297–1311, Jul. 2016, doi: https://doi.org/10.1080/00207543.2016.1201604

[6] M. Dayarathna and S. Perera, “Recent Advancements in Event Processing,” *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–36, Jun. 2018, doi: https://doi.org/10.1145/3170432.

[7] I. Konovalenko and A. Ludwig, “Event processing in supply chain management – The status quo and research outlook,” *Computers in Industry*, vol. 105, pp. 229–249, Feb. 2019, doi: https://doi.org/10.1016/j.compind.2018.12.009.

[8] Z. Milosevic, W. Chen, A. Berry, and F. A. Rabhi, “Real-Time Analytics,” *Big Data*, pp. 39–61, 2016, doi: https://doi.org/10.1016/b978-0-12-805394-2.00002-7.

[9] M. Brambilla and P. Fraternali, "IFML language design, execution, and integration," *Elsevier eBooks*, pp. 359–380, Jan. 2015, doi: https://doi.org/10.1016/b978-0-12-800108-0.00012-6.
[10] M. G. Valls, A. Dubey, and V. Botti, "Introducing the new paradigm of Social Dispersed Computing: Applications, Technologies and Challenges," *Journal of Systems Architecture*, vol. 91, pp. 83–102, Nov. 2018, doi: https://doi.org/10.1016/j.sysarc.2018.05.007.
[11] Z. Zheng, P. Wang, J. Liu, and S. Sun, "Real-Time Big Data Processing Framework: Challenges and Solutions," *Appl. Math. Inf. Sci*, vol. 9, no. 6, pp. 3169–3190, 2015, doi: https://doi.org/10.12785/amis/090646.
[12] M. Emmi, P. Ganty, R. Majumdar, and F. R. Velardo, "Analysis of Asynchronous Programs with Event-Based Synchronization," *Lecture notes in computer science*, pp. 535–559, Jan. 2015, doi: https://doi.org/10.1007/978-3-662-46669-8_22.
[13] A. Katsifodimos and M. Fragkoulis, " Operational Stream Processing: Towards Scalable and Consistent Event-Driven pplications," 2019, doi: https://doi.org/10.5441/002/edbt.2019.86.
[14] S. Ibanez, G. Antichi, G. Brebner, and N. McKeown, "Event-Driven Packet Processing," Nov. 2019, doi: https://doi.org/10.1145/3365609.3365848.
[15] R. J. Wickham, "Secondary Analysis Research - JADPRO," *NCBI*, 2019, doi: https://doi.org/10.6004%2Fjadpro.2019.10.4.7.
[16] G. V. Machado, Í. Cunha, A. C. M. Pereira, and L. B. Oliveira, "DOD-ETL: distributed on-demand ETL for near real-time business intelligence," *Journal of Internet Services and Applications*, vol. 10, no. 1, Nov. 2019, doi: https://doi.org/10.1186/s13174-019-0121-z.
[17] A. Hassani and S. A. Gahnouchi, "A framework for Business Process Data Management based on Big Data Approach," *Procedia Computer Science*, vol. 121, pp. 740–747, 2017, doi: https://doi.org/10.1016/j.procs.2017.11.096.
[18] A. Sherif, "Business intelligence penetration by industry 2018," *Statista*, 2018. https://www.statista.com/statistics/873143/business-intelligence-penetration-by-industry/ (accessed Jul. 18, 2019).
[19] G. Liu, W. Zhu, C. Saunders, F. Gao, and Y. Yu, "Real-time Complex Event Processing and Analytics for Smart Grid," *Procedia Computer Science*, vol. 61, pp. 113–119, 2015, doi: https://doi.org/10.1016/j.procs.2015.09.169.
[20] Semantic Scholar, "Event Driven Architecture in software development projects," *Semantic Scholar*, 2016. https://www.semanticscholar.org/paper/Event-Driven-Architecture-in-software-development/f98ae457d75b0b431c6f17a3f396716c947a4d99 (accessed Jul. 18, 2019).
[21] O.-A. Schipor, R.-D. Vatavu, and J. Vanderdonckt, "Euphoria: A Scalable, event-driven architecture for designing interactions across heterogeneous devices in smart environments," *Information and Software Technology*, vol. 109, pp. 43–59, May 2019, doi: https://doi.org/10.1016/j.infsof.2019.01.006.
[22] A. Theorin *et al.*, "An Event-Driven Manufacturing Information System Architecture," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 547–554, 2015, doi: https://doi.org/10.1016/j.ifacol.2015.06.138.
[23] M. Ayoubi, A. R. Shinwari, and M. S. Khan, "An Event-Driven Service Oriented Architecture Approach for E-Governance Systems," *Kardan journal of engineering and technology*, Dec. 2019, doi: https://doi.org/10.31841/kjet.2021.1.
[24] Apache Flink, "Introducing Stream Windows in Apache Flink," *Apache Flink*, Dec. 04, 2015. https://flink.apache.org/2015/12/04/introducing-stream-windows-in-apache-flink/ (accessed Jul. 18, 2019).
[25] N. Alshuqayran, N. Ali, and R. Evans, "A Systematic Mapping Study in Microservice Architecture," *research.brighton.ac.uk*, pp. 44–51, Nov. 2016, doi: https://doi.org/10.1109/SOCA.2016.15.