Conformance Checking on Out-of-Order Streams (Extended Abstract)

Kristo Raun
Institute of Computer Science
University of Tartu
Tartu, Estonia
kristo.raun@ut.ee

***Abstract*—Conformance checking is a subfield of process mining that deals with comparing process models to event logs. Usually, it is done in an offline fashion. With the increased connectivity to the internet and ubiquity of data generators, increased attention is given to acquiring fast and reliable conformance checking results in an online setting. In return, a number of approaches have been developed to enable online conformance checking based on so-called prefix alignments. However, the current approaches assume that the events of process execution are arriving in order on the streams. Unfortunately, due to the complexity of networks and the distributed nature of processing, out-of-order arrival is the common case.**

The main focus of this PhD is to develop performant and self-correcting methods for online conformance checking which can handle out-of-order event arrival. Additionally, the work will include finding methods which will best utilize approximate techniques while minimizing the impact on the accuracy. Finally, to be applicable in real life settings, the research results will be implemented on top of a distributed stream processing system, e.g. Apache Spark.

***Index Terms*—conformance checking, streaming data, out of order, big data**

I. Problem Statement

The last decade has seen a major shift in computing. With more systems and more data points, the volume of data available has exploded. Traditionally, most data processing and data analysis was done in nightly batches, but now many organizations have found use cases for analyzing data in a more continuous fashion, by utilizing streaming data. Various streaming engines have been developed and seen rapid adoption in the industry, among others Apache Spark, Apache Kafka, Apache Flink, Apache Storm.

Recently, *online* conformance checking has drawn the attention of researchers and practitioners, with a surge of relevant publications in the last few years, e.g. [1], [2]. Due to the novelty of the research field, some areas have remained understudied and many challenges, which can surface in truly streaming systems, have seen little or no attention in the online conformance checking research. Particularly, handling out-of-order events is of major relevance in streaming applications. To the best of knowledge, it has so far had only one mention in relevant literature [3]. Out-of-order events are likely to happen in cases where multiple systems are responsible for various parts of the process, which is quite common in modern software architecture. Events may arrive out of order to the processing engine due to reasons such as delays in the network or system design. For example, a business critical system may publish events to an analysis service only after certain intervals. Looking at Fig. 1, event 5 is the final event in the process, and the processing engine has just received event 5. It could falsely conclude that the trace is not conforming: firstly, events 2 and 3 are not in the correct order; secondly, event 4 has not yet arrived. However, the events are merely arriving too late because of the system handling events 2 and 4.

The goal of this PhD project is to develop performant and self-correcting methods for online conformance checking which are able to handle out-of-order event arrival. This goal is going to be fulfilled by answering the following research questions.

RQ1: *How can we develop a reactive, performant and self-correcting online conformance checking system?*

The analysis system needs to be *reactive*, i.e. constantly looking for new events and ready to process them. The system also needs to be *performant*, as the processing would be expected to happen in near real time in actual settings. Furthermore, due to possible issues in data streaming, the system needs to be *self-correcting*: the arrival of an event should trigger a correction on the conformance, specifically in case a discrepancy exists between what the system knew before the arrival of the event and after.

RQ2: *How to represent process behavior in succinct and efficiently searchable data structures?*

State-of-the-art conformance checking is costly and this is even more true in an online setting. The method needs to be able to traverse through a trace and compare it to a model, while also taking into account potential discrepancies due to out-of-order events that need to be corrected. This implies potentially a very large data structure which needs to be kept in memory of the processing engine. To deal with the potential memory issue, the underlying data structure has to be *succinct*. To be performant and have low latency, the data structure should be *efficiently searchable*.

II. State of the Art

Online Conformance Checking A general framework for online conformance checking was introduced in [4], together with a method which allows conformance checking on event streams, instead of static event logs. Measures for monitoring

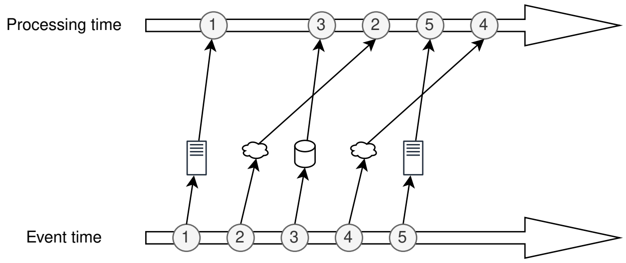

Fig. 1. Streaming events arriving out of order due to different source systems.

the completeness of the process execution were introduced in [5], together with the notion of warm starting: assuming some activities have been executed but not observed. Some scholars have looked at utilizing prefix-alignments [1], [2], as calculating the full alignment for a trace in an online setting may overestimate the discrepancy between the trace and the model.

As previously discussed, it seems that no research in online conformance checking has fully acknowledged the notion of out-of-order data arrival in an event stream. Thus, handling out-of-order event streams seems to be an important contribution to the advancement of online conformance checking.

Approximate Conformance Checking Computational complexity of alignments has led to research in approximation methods. The work in [6] defines approximate alignments as such that moves can happen in multisets of activities. Several methods [6], [7] rely on ILP (Integer Linear Programming), achieving approximate alignments that have a relatively small error compared to the optimal alignment, indicating their viability in practice.

A part of approximation research has recently shifted to selecting a sample of the event log [8], [9]. Approximation methods in [9], [10] have shown significant improvement in computation time with marginal error. Recently, in [11] we used a trie data structure to further lower the computation time of approximate alignments while introducing a small additional error.

Using tries can be one answer to the second research question. Still, there are many more methods which have not had too much research attention for efficient conformance checking, such as suffix arrays and novel string matching methods.

III. Research Plan

For addressing latency and in particular the problem of handling out-of-order events, various solutions exist which could be studied in the context of conformance checking, such as dynamic buffer sizing [12]. This could be further extended with statistical or machine learning methods for discovering the optimal behaviour of the buffer based on previous patterns in the event stream.

An important part of the research will be to find efficient data structures which can be utilized in an online setting. The methods should strike a balance between memory consumption, accuracy of alignment and the latency to processing engine. Addressing the memory-accuracy trade-off could be studied from the field of genomics, where large strings of data need to be compared. In addition to prefix-alignments, other relevant notions from the fields of bibliometrics and genomics could be considered as well, like suffix arrays and substrings [13]. We have already found tries useful for approximating alignments, as discussed in [11], but the work could be further developed to work in an online context.

In order to have reproducibility and wider academic and industry impact, any methods developed within this research should be made available through an open-source framework, such as ProM or PM4Py.

References

[1] D. Schuster and G. J. Kolhof, "Scalable online conformance checking using incremental prefix-alignment computation," in *Service-Oriented Computing – ICSOC 2020 Workshops*, H. Hacid, F. Outay, H.-y. Paik, A. Alloum, M. Petrocchi, M. R. Bouadjenek, A. Beheshti, X. Liu, and A. Maaradji, Eds. Cham: Springer International Publishing, 2021, pp. 379–394.

[2] S. J. van Zelst, A. Bolt, M. Hassani, B. F. van Dongen, and W. M. van der Aalst, "Online conformance checking: relating event streams to process models using prefix-alignments," *International Journal of Data Science and Analytics*, vol. 8, no. 3, pp. 269–284, 2019.

[3] A. Awad, M. Weidlich, and S. Sakr, "Process mining over unordered event streams," in *2020 2nd International Conference on Process Mining (ICPM)*, 2020, pp. 81–88.

[4] A. Burattin and J. Carmona, "A framework for online conformance checking," in *Business Process Management Workshops*, E. Teniente and M. Weidlich, Eds. Cham: Springer International Publishing, 2018, pp. 165–177.

[5] A. Burattin, S. J. van Zelst, A. Armas-Cervantes, B. F. van Dongen, and J. Carmona, "Online conformance checking using behavioural patterns," in *Business Process Management*, M. Weske, M. Montali, I. Weber, and J. vom Brocke, Eds. Cham: Springer International Publishing, 2018, pp. 250–267.

[6] F. Taymouri and J. Carmona, "A recursive paradigm for aligning observed behavior of large structured process models," in *Business Process Management*, M. La Rosa, P. Loos, and O. Pastor, Eds. Cham: Springer International Publishing, 2016, pp. 197–214.

[7] B. van Dongen, J. Carmona, T. Chatain, and F. Taymouri, "Aligning modeled and observed behavior: A compromise between computation complexity and quality," in *Advanced Information Systems Engineering*, E. Dubois and K. Pohl, Eds. Cham: Springer International Publishing, 2017, pp. 94–109.

[8] M. Bauer, H. van der Aa, and M. Weidlich, "Estimating process conformance by trace sampling and result approximation," in *Business Process Management*, T. Hildebrandt, B. F. van Dongen, M. Röglinger, and J. Mendling, Eds. Cham: Springer International Publishing, 2019, pp. 179–197.

[9] M. FaniSani, S. J. van Zelst, and W. M. P. vander Aalst, "Conformance checking approximation using subset selection and edit distance," in *Advanced Information Systems Engineering*, S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant, Eds. Cham: Springer International Publishing, 2020, pp. 234–251.

[10] M. F. Sani, J. J. G. Gonzalez, S. J. van Zelst, and W. M. van der Aalst, "Conformance checking approximation using simulation," in *2020 2nd International Conference on Process Mining (ICPM)*, 2020, pp. 105–112.

[11] A. Awad, K. Raun, and M. Weidlich, "Efficient approximate conformance checking using trie data structures," in *2021 3rd International Conference on Process Mining (ICPM)*, 2021, In press.

[12] W. Weiss, V. J. E. Jiménez, and H. Zeiner, "Dynamic buffer sizing for out-of-order event compensation for time-sensitive applications," *ACM Trans. Sen. Netw.*, vol. 17, no. 1, Sep. 2020. [Online]. Available: https://doi.org/10.1145/3410403

[13] Y. Chen and H. H. Nguyen, "On the string matching with ¡i¿k¡/i¿ differences in dna databases," *Proc. VLDB Endow.*, vol. 14, no. 6, p. 903915, Feb. 2021. [Online]. Available: https://doi.org/10.14778/3447689.3447695