Proactive Streaming Analytics at Scale: A Journey from the State-of-the-art to a Production Platform

Nikos Giatrakos
ngiatrakos@softnet.tuc.gr
Technical University of Crete
Chania, Greece

Elias Alevizos
alevizos.elias@iit.demokritos.gr
National Centre For Scientific
Research Demokritos
Agia Paraskevi, Greece

Antonios Deligiannakis
adeli@softnet.tuc.gr
Technical University of Crete
Chania, Greece

Ralf Klinkenberg
rklinkenberg@altair.com
Altair Engineering GmbH
Dortmund, Germany

Alexander Artikis
a.artikis@iit.demokritos.gr
National Centre For Scientific
Research Demokritos
Agia Paraskevi, Greece

ABSTRACT

Proactive streaming analytics continuously extract real-time business value from massive data that stream in data centers or clouds. This requires (a) to process the data while they are still in motion; (b) to scale the processing to multiple machines, often over various, dispersed computer clusters, with diverse Big Data technologies; and (c) to forecast complex business events for proactive decision-making. Combining the necessary facilities for proactive streaming analytics at scale entails: (I) deep knowledge of the relevant state-of-the-art, (II) cherry-picking cutting edge research outcomes based on desired features and with the prospect of building interoperable components, and (III) building components and deploying them into a holistic architecture within a real-world platform. In this tutorial, we drive the audience through the whole journey from (I) to (III), delivering cutting edge research into a commercial analytics platform, for which we provide a hands-on experience.

CCS CONCEPTS

• **Information systems** → **Online analytical processing engines**; **Stream management**; **MapReduce-based systems**; • **Software and its engineering** → *Software design engineering*.

KEYWORDS

big streaming data, synopses, complex event forecasting, optimizer

ACM Reference Format:
Nikos Giatrakos, Elias Alevizos, Antonios Deligiannakis, Ralf Klinkenberg, and Alexander Artikis. 2023. Proactive Streaming Analytics at Scale: A Journey from the State-of-the-art to a Production Platform. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583780.3615293

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615293

1 BACKGROUND – USE CASES – OBJECTIVES

At an increasing rate, numerous industrial and scientific institutions face business requirements for real-time, proactive analytics to derive actionable items and timely support decision-making procedures. In the stock market domain, proactive analytics enable timely reaction to opportunities or risks. In maritime surveillance, voluminous position streams of thousands of vessels, satellite images or acoustic signal streams are fused to predict illegal activities [12].

Big Data platforms such as Apache Flink or Apache Spark have designed streaming APIs to facilitate hotizontally scaling-out, i.e., parallelizing, the computation of streaming analytics tasks to a number of Virtual Machines (VM) available in corporate computer clusters or the cloud. Useful as these facilities may be, they only focus on a narrow part of the challenges that proactive analytics workflows need to encounter in streaming settings.

First, Big Data platforms currently provide none or suboptimal support for proactive streaming analytics tasks engaging Machine Learning (ML) operators. The major ML APIs they provide, such as MLlib or FlinkML, do not focus on parallel implementations of streaming algorithms. The same holds for proactive analytics via forecasting complex business events [21].

Second, Big Data platforms by design focus only on horizontal scalability as described above. To materialize additional types of scalability, such as vertical and federated scalability, stream summarization techniques come in handy. Surprisingly, Big Data platforms have no libraries or APIs dedicated to stream summaries.

Third, Big Data technologies are significantly fragmented. Delivering continuous proactive analytics at scale requires optimizing the execution of workflows over a variety of Big Data platforms at a number of, potentially geo-dispersed, clusters or clouds.

Motivated by the above, the goals and objectives of this tutorial are: (1) to provide a comprehensive study of the state-of-the-art on (a) distributed complex event forecasting, (b) stream summarization over distributed settings, (c) inter-cluster and cross-platform optimization, (2) to reason about cherry-picking state-of-the-art techniques to build respective architectural components, (3) to provide a hands-on experience on integrating these components into a generic architecture, incorporated in a commercial analytics platform, used in real-world use cases. All components used in this tutorial are provided open-source by the authors (footnotes 1 to 4).

2 CONTENT AND ORGANIZATION

Our presentation covers the topics cited next, complemented with analytical examples, insights and lessons learned:

Introduction

- Motivation, Use Cases and Challenges

(Geo-)Distributed Proactive Stream Analytics

- The Case for Stream Synopses [4, 17, 39, 49]
- Complex Event Forecasting (CEF) [20, 21, 34, 46]
- Inter-/Intra-cluster Optimization [1, 5, 9, 16, 28, 35, 48, 53]

Cherry-picking Research & Building Components

- Synopses-as-a-Service Paradigm [11, 39]
- Options for Distributed CEF Architecture [22, 33]
- Integrating Multi-cluster, Cross-platform & Adaptive Optimization Concepts [13, 25, 27, 35, 36]

Hands-on Experience

- System Architecture
- Workflow Construction & Execution

Open Issues & Takeaways

3 STATE OF THE ART

3.1 Stream Summarization

Synopses provide approximate answers, with accuracy guarantees, to popular analytic operators [2, 24, 32], simultaneously reducing processing and memory loads. In that, they can boost horizontal scalability provided by Big Data platforms, at scale. However, in (geo-)distributed streaming analytics there are two additional types of scalability that are not covered by Big Data platforms whatsoever. Vertical scalability, i.e., scaling the computation with the number of processed streams, is also a necessity. Additionally, federated scalability involves communication reduction in geo-distributed streaming settings composed of multiple, potentially geo-dispersed computer clusters. Here, synopses aid in scaling the computation by allowing the communication of compact data summaries.

Apache DataSketches [4] and Stream-lib [50] are software libraries of stochastic streaming algorithms and summarization techniques. Although, these libraries are detached from parallelization aspects, their functionality can be used in Big Data platform programs. However, horizontal scalability configuration is left to the programmer. SnappyData's [49] stream processing is based on Spark and incorporates a limited set of synopses serving simple `SUM`, `COUNT` and `AVG` queries. Similarly, StreamApprox [17] offers only sampling as a pipeline operator. Thus, these are deprived of vertical scalability features and federated scalability provisions. The prominent work of Condor [39] elegantly optimizes the parallel computation of stream summaries, still neglecting aspects of horizontal scalability, as well as vertical and federated scalability. Table 1 summarizes our above discussion.

3.2 Complex Event Forecasting

Complex Event Recognition (CER) is one of the technologies with increased popularity [3, 34] when the main goal is to first learn and then detect interesting activity patterns occurring within a stream of events, coming from on-field devices or sensors. Complex Events must be detected with minimal latency. As a result, a significant body of work has been devoted to computational optimization issues. Less attention has been paid to forecasting event patterns [34], despite the fact that forecasting has attracted considerable attention in various related research areas, such as time-series forecasting [18], sequence prediction [19, 23, 37, 38], temporal mining [14, 15, 40, 44] and process mining [8]. Consider, for example, credit card fraud management in the financial domain [7], where the detection of suspicious activity patterns of credit cards must occur with minimal latency that is in the order of a few milliseconds. The decision margin is extremely narrow. Being able to forecast that a certain sequence of transactions is very likely to be a fraudulent pattern provides wider margins both for decision and for action. For example, a processing system might decide to devote more resources and higher priority to those suspicious patterns to ensure that the latency requirement will be satisfied. The need for Complex Event Forecasting (CEF) has been acknowledged though, as evidenced by several conceptual proposals [6, 30, 31]. The few previous concrete attempts at CEF [43, 45, 47] have various limitations. Either they do not even target complex event forecasting, focusing instead on simple event forecasting, or they employ relatively simple probabilistic models that cannot uncover deep dependencies.

3.3 Inter-cluster/Cross-platform Optimization

A crucial feature for optimizing streaming workflows involves the ability to adapt a currently deployed workflow execution plan at runtime. This is due to the high volatility of streaming workloads that can rapidly render a previously preferable execution plan to severely suboptimal. Cross-platform systems have evolved [5, 9, 16, 28, 42], some incorporating their own optimizers. Nonetheless, their focus is on batch, instead of streaming settings [26] and rarely consider runtime adaptation scenarios [1]. Even systems that support stream processing, are restricted to a single streaming platform [16, 29]. Few stream processing systems [1, 48, 52] have touched upon aspects of adaptive re-scaling of streaming workflows, focusing on a single engine as well. Table 2 summarizes the stream processing and optimization capacity of major relevant frameworks. Optimizing in geo-distributed, cross-platform streaming settings is a totally different business because optimization must support in a unified way, across platforms and throughout the network of clusters, the following features: (a) adaptation and migration of a deployed workflow/query plan at runtime, (b) instant new execution plan generation for arbitrarily complex workflows and networks, and (c) lightweight and incremental performance modelling.

4 CHERRY-PICKS FOR S/W COMPONENTS

Synopses Component: To tackle the gaps left by the aforementioned efforts, we illustrate a new synopses maintenance paradigm, namely Synopses Data Engine-as-a-Service (SDEaaS)[1] [10, 11]. SDEaaS synthesizes the software technology concepts used in Stream-lib [50] and includes a rich set of synopses integrating [4, 17, 49]. It better handles CPU core allocation compared to these works mainly because synopses can be maintained by a single running job in one or more clusters. SDEaaS accounts for all three types of required scalability and can accept on-the-fly requests for

[1] https://sdeaas.github.io

Scalability→ Synopses Approach↓	Horizontal	Vertical	Federated
DataSketch [4]	❑ (Spark Integration)	✘	✘
Stream-lib [50]	✘	✘	✘
StreamApprox [17]	❑ (Stratified Sampling)	✘	✘
SnappyData [49]	❑ (Simple Aggregates)	✘	✘
Condor [39]	✔	✘	✘
SDaaS [11]	✔	✔	✔

Table 1: Stream Summarization & Scalability

Features→ System↓	Stream Support	Synopses Support	Optimization			Proactive Analytics	
			Cross-Platform	Multi-Cluster	Runtime Adaptation	Parallel/ Distributed ML	CEF
Musketeer [28]	✘	✘	✔	✘	✘	✘	✘
IReS [5]	✘	✘	✔	✘	✘	✘	✘
BigDawg [9]	❑ (S-Store)	✘	✔	✘	✘	✘	✘
Rheem [16]	❑ (JavaStreams)	✘	✔	✘	✘	✔	✘
INforE [35]	✔	✔	✔	✔	✘	✔	✘
SheerMP [25]	✔	✔	✔	✔	✔	✔	✘
Wayeb [21, 46]	✔	✘	✘	✘	✘	✔	✔

Table 2: Required Features for Proactive Streaming Analytics at Scale

plugging-in new synopses techniques into its library and/or maintaining synopses on demand. It also allows for ad-hoc or continues queries on maintained summaries even in cross-platform scenarios.
CEF Component: Tackling shortcomings of previous CEF and CER approaches, we illustrate a CEF framework that is both formal and easy to use, thus avoiding confusion about how patterns should be written and which operators are allowed [34]. Our framework (Wayeb[2]) is formal, compositional and as easy to use as simply writing regular expressions. The user declaratively defines a pattern and provides a training data stream. Wayeb can also uncover deep probabilistic dependencies in a stream by using a variable-order Markov model. Additionally, Wayeb can perform various types of forecasting, both for simple events (i.e., predicting what the next input event might be) and Complex Events (events defined through a pattern). Thus, it overcomes restrictions of previous methods.
Optimizer: We illustrate an optimizer, namely SheerMP [25] [3], that unifies the optimization of multi-cluster, cross-platform streaming workflows by synthesizing and significantly extending virtues of prior work, providing: (i) both rapid, best-effort optimization approaches [41, 53] and more time-consuming algorithms guaranteeing to devise optimal plans [27], (ii) seamless runtime adaptation/job migration [1], (ii) computationally inexpensive, incremental cost model construction [36], (iii) lightweight statistics collection [51].

5 HANDS-ON RAPIDMINER STUDIO

The constructed components are generic enough to be used in isolation, without tying them to a specific platform. However, to boost proactive analytics at scale one needs to be able to concurrently exploit their advanced features in application workflows. To achieve that, we incorporate the involved components into a commercial platform, namely RapidMiner studio. The streaming extension we provide to the Studio[4] enables analysts to easily design proactive streaming analytics workflows without coding. This is achieved by encapsulating the constructed components as operators represented by boxes. The Optimizer Component extends RapidMiner Studio by a so-called "Nest" operator. The Nest operator is a sub-process operator, which means that families of operators (from the Synopses Data Engine, CEF Components and stream transformations provided by Big Data platforms) can be placed inside it. Having placed a Nest operator in a workflow, the user can double click on it and then encapsulate other operators/boxes. This is done in a GUI via drag and drop actions and operators can be connected by drawing arrows to define the data flow. We use a dictionary to map platform-agnostic operators of designed workflows to physical operators for the supported platforms and networked clusters. Apache Kafka connection objects are used to fuse operator input/outputs. The hands-on experience on the streaming extension of RapidMiner Studio comes in the form of designing and parameterizing workflows for application scenarios mentioned in Section 1. Sample Videos: Maritime Use Case & Forecasting (https://youtu.be/q2wxlgLjjiQ), Financial Use Case & Optimization: (https://youtu.be/68zyJNoEjiU)

6 OPEN ISSUES

Stream Summarization. The approach of SDEaaS [11] is, in principle, complementary to Condor [39]. On one hand, SDEaaS fosters simple, per stream or round-robin parallelization schemes. On the other hand, Condor lacks support for vertical and federated scalability. Combining Condor with SDEaaS, preserving their advantages, is a non-trivial task future research should encounter. Proving their portability in other Big Data platforms also remains an open issue.
Complex Event Forecasting. CEF currently works by assuming the complex events are defined via patterns. However, often analysts may have labels for complex events without having a definition for them. How one could forecast such complex events, even in the absence of event definitions (e.g., by first extracting those definitions from the existing labels and then performing CEF as usual)? Moreover, CEF could be used for pattern-driven lossless stream compression, to minimize the communication cost, which is a severe bottleneck for geo-distributed CER [34]. The probabilistic model that CEF constructs should be pushed down to the event sources, to compress each individual stream before transmitting to a central source. This is a novel compression concept not considered so far.
Optimization. A major system research direction involves the fact that job migration from one Big Data platform to another is not supported. Even unifying programming models such as Apache Beam, currently establish no clear equivalence between the streaming APIs of popular Big Data platforms. Additionally, SheerMP [25] employs an operator-based scheme while Jarvis [13] optimizes migration via clever data partitioning over multiple sources, but a single processing cluster. Replicating operators, simultaneously partitioning their inputs over multiple clusters and platforms remains an open issue. Finally, SheerMP does not optimize CEF operators, but the Wayeb Optimizer [46] specializes the generic optimization approach of SheerMP. Integrating CEF optimization support to SheerMP would be an important extension left for future work.

[2]https://github.com/ElAlev/Wayeb
[3]https://bitbucket.org/infore_research_project/optimizer-release/
[4]https://bitbucket.org/infore_research_project/rapidminer-extension-streaming-release

7 PRESENTERS

Nikos Giatrakos is an Assistant Professor at the School of ECE, Technical University of Crete. His research focuses on software architectures, algorithms and systems for Big Data Management including Big streaming Data, Decentralized Big Data Processing, Stream Summarization and Complex Event Processing.

Elias Alevizos is a post-doctoral researcher at NCSR "Demokritos". His research interests lie in the fields of Data Science, Artificial Intelligence and Complex Event Recognition/Forecasting.

Antonios Deligiannakis is a Professor at the School of ECE, Technical University of Crete. His research interests are in the area of Big Data Analytics over data streams, including distributed data processing, complex event processing in large scale systems, approximate query processing and large scale data mining.

Ralf Klinkenberg, founder and head of research at RapidMiner and senior director of data science research at Altair, is a data-driven entrepreneur with more than 30 years of experience in ML, AI and advanced data analytics research, software development and consulting. In 2008 he won the European Open Source Business Award and in 2016 the European Data Innovator Award. Today, RapidMiner has more than 1 Mio. registered users in more than 150 countries world-wide and is one of the most widely used predictive analytics platforms.

Alexander Artikis is an Associate Professor at the University of Piraeus, and a Research Associate at NCSR 'Demokritos', leading the Complex Event Recognition group. He has more than 100 publications in the fields of Artificial Intelligence and Data Science. According to Google Scholar his h-index is 36. Alexander has given tutorials in various conferences, such as VLDB, IJCAI and KR, and has co-organised the Dagstuhl seminar on the Foundations of Composite Event Recognition.

ACKNOWLEDGEMENTS

This work was supported by the EU projects CREXDATA (for E. Alevizos, A. Deligiannakis, R. Klinkenberg and A. Artikis) under Horizon Europe agreement No. 101092749 and EVENFLOW (for N. Giatrakos) under Horizon Europe agreement No. 101070430.

REFERENCES

[1] V. Cardellini et al. 2022. Runtime Adaptation of Data Stream Processing Systems: The State of the Art. *ACM Comput. Surv.* 54, 11s, Article 237 (sep 2022), 36 pages.
[2] G. Cormode and K. Yi. 2020. *Small Summaries for Big Data.* Cambridge University Press.
[3] G. Cugola and A. Margara. 2012. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.* 44, 3 (2012), 15:1–15:62.
[4] Apache DataSketches. 2020. https://datasketches.github.io/.
[5] K. Doka et al. 2015. IReS: Intelligent, Multi-Engine Resource Scheduler for Big Data Analytics Workflows. In *SIGMOD*.
[6] Y. Engel and O. Etzion. 2011. Towards proactive event-driven computing. In *DEBS*.
[7] A. Artikis et al. 2017. A Prototype for Credit Card Fraud Management: Industry Paper. In *DEBS*.
[8] A. Eduardo Márquez-Chamorro et al. 2018. Predictive Monitoring of Business Processes: A Survey. *IEEE Trans. Services Computing* 11, 6 (2018), 962–977.
[9] A. J. Elmore et al. 2015. A Demonstration of the BigDAWG Polystore System. *Proc. VLDB Endow.* 8, 12 (2015).
[10] A. Kontaxakis et al. 2020. A Synopses Data Engine for Interactive Extreme-Scale Analytics. In *CIKM*.
[11] A. Kontaxakis et al. 2023. And synopses for all: A synopses data engine for extreme scale analytics-as-a-service. *Information Systems* 116 (2023), 102221.
[12] A. Milios et al. 2019. Automatic Fusion of Satellite Imagery and AIS data for Vessel Detection. In *FUSION*.
[13] A. Sandur et al. 2022. Jarvis: Large-scale Server Monitoring with Adaptive Near-data Processing. In *ICDE*.
[14] Chung-Wen Cho et al. 2011. On-line rule matching for event prediction. *VLDB J.* 20, 3 (2011), 303–334.
[15] C. Zhou et al. 2015. A pattern based predictor for event streams. *Expert Syst. Appl.* 42, 23 (2015), 9294–9306.
[16] D. Agrawal et al. 2018. RHEEM: Enabling Cross-Platform Data Processing - May The Big Data Be With You! -. *Proc. VLDB Endow.* 11, 11 (2018).
[17] D. L. Quoc et al. 2017. StreamApprox: approximate computing for stream analytics. In *Middleware*.
[18] D. Montgomery et al. 2015. *Introduction to time series analysis and forecasting*. John Wiley & Sons.
[19] D. Ron et al. 1996. The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length. *Machine Learning* 25, 2-3 (1996), 117–149.
[20] E. Alevizos et al. 2018. Wayeb: a Tool for Complex Event Forecasting. In *LPAR*.
[21] E. Alevizos et al. 2022. Complex event forecasting with prediction suffix trees. *VLDB J.* 31, 1 (2022).
[22] E. Ntoulias et al. 2021. Online trajectory analysis with scalable event recognition. In *EDBT/ICDT (CEUR Workshop Proceedings)*.
[23] F. M. J. Willems et al. 1995. The context-tree weighting method: basic properties. *IEEE Trans. Information Theory* 41, 3 (1995), 653–664.
[24] G. Cormode et al. 2012. Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches. *Foundations and Trends in Databases* 4, 1-3 (2012).
[25] G. Stamatakis et al. 2022. SheerMP: Optimized Streaming Analytics-as-a-Service over Multi-site and Multi-platform Settings. In *EDBT*.
[26] H. Herodotou et al. 2020. A Survey on Automatic Parameter Tuning for Big Data Processing Systems. *ACM Comput. Surv.* 53, 2 (2020).
[27] I. Flouris et al. 2020. Network-wide complex event processing over geographically distributed data sources. *Inf. Syst.* 88 (2020).
[28] I. Gog et al. 2015. Musketeer: all for one, one for all in data processing systems. In *EuroSys*.
[29] J. Meehan et al. 2016. Integrating real-time and batch processing in a polystore. In *HPEC*.
[30] L. J. Fülöp et al. 2012. Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics. In *BCI*.
[31] M. Christ et al. 2016. Integrating Predictive Analytics into Complex Event Processing by Using Conditional Density Estimations. In *EDOC Workshops*.
[32] M. Garofalakis et al. 2016. Data Stream Management: A Brave New World. In *Data Stream Management - Processing High-Speed Data Streams*.
[33] M. Vodas et al. 2021. Online Distributed Maritime Event Detection & Forecasting over Big Vessel Tracking Data. In *IEEE Big Data*.
[34] N. Giatrakos et al. 2020. Complex event recognition in the Big Data era: a survey. *VLDB J.* 29, 1 (2020), 313–352.
[35] N. Giatrakos et al. 2020. INforE: Interactive Cross-platform Analytics for Everyone. In *CIKM*.
[36] O. Alipourfard et al. 2017. CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics. In *NSDI*.
[37] P. Bühlmann et al. 1999. Variable length Markov chains. *The Annals of Statistics* 27, 2 (1999), 480–513.
[38] R. Begleiter et al. 2004. On Prediction Using Variable Order Markov Models. *J. Artif. Intell. Res.* 22 (2004), 385–421.
[39] R. P. Lemaitre et al. 2021. In the Land of Data Streams where Synopses are Missing, One Framework to Bring Them All. *Proc. VLDB Endow.* 14, 10 (2021).
[40] R Vilalta et al. 2002. Predicting Rare Events In Temporal Domains. In *ICDM*. IEEE Computer Society, 474–481.
[41] S. Beamer et al. 2013. Distributed Memory Breadth-First Search Revisited: Enabling Bottom-Up Search. In *IPDPSW*.
[42] S. Chatterjee et al. 2021. Cosine: A Cloud-Cost Optimized Self-Designing Key-Value Storage Engine. *PVLDB* 15, 1 (2021).
[43] S. Gillani et al. 2017. Pi-CEP: Predictive Complex Event Processing Using Range Queries over Historical Pattern Space. In *ICDM Workshops*. IEEE Computer Society, 1166–1171.
[44] S. Laxman et al. 2008. Stream prediction using a generative model based on frequent episodes in event sequences. In *KDD*. ACM, 453–461.
[45] V. Muthusamy et al. 2010. Predictive publish/subscribe matching. In *DEBS*.
[46] V. Stavropoulos et al. 2022. Optimizing complex event forecasting. In *DEBS*.
[47] Y. Li et al. 2020. Data Stream Event Prediction Based on Timing Knowledge and State Transitions. *Proceedings of the VLDB Endowment* 13, 10 (2020).
[48] A. Floratou et al. 2017. Dhalion: Self-Regulating Stream Processing in Heron. *PVLDB* 10, 12 (2017).
[49] B. Mozafari. 2019. SnappyData. In *Encyclopedia of Big Data Technologies*.
[50] Stream-lib. 2019. Stream-lib. https://github.com/addthis/stream-lib.
[51] G. van Dongen et al. 2020. Evaluation of Stream Processing Frameworks. *IEEE Transactions on Parallel and Distributed Systems* 31, 8 (2020), 1845–1858.
[52] S. Venkataraman et al. 2017. Drizzle: Fast and Adaptable Stream Processing at Scale. In *SoSP*.
[53] F. Waas and A. Pellenkoft. 2000. Join Order Selection - Good Enough Is Easy *(BNCOD 17)*. 51-67.