IT Operations Analytics: Root Cause Analysis via Complex Event Processing

Martin Drasar*, Tomas Jirsik*†
*Institute of Computer Science, †Faculty of Informatics
Masaryk University, Brno, Czech Republic
{drasar,jirsik}@ics.muni.cz

***Abstract*—IT operation analytics (ITOA) is used for discovering complex patterns in data from IT systems. The analytics process still includes a significant portion of human interaction which makes the analysis costly and error-prone. Human operators need to formulate queries over the collected data to identify the complex patterns. Since the queries describe complex relations, the queries are usually multilevel, perplexing, and complicated to create. For the querying the complex relations, complex event processing methods are successfully used in other domains. In this paper, we demonstrate an application of the complex event processing principles in the ITOA domain. We adjust T-Rex complex event processing engine and improve TESLA event processing language to suit for ITOA tasks. Our demonstration includes two real-world use-cases. We show the utilization of the complex event processing for root cause analysis and demonstrate the natural formulation of complex queries that results in the reduction of the volume of the required human interaction.**

I. Introduction

Network visibility and application performance analysis have improved significantly in recent years as novel technologies for network and application monitoring are continuously introduced by industry vendors. Network operators and information systems managers are provided with a large volume of data on network behavior and application performance that needs to be processed and understood to retrieve relevant information. Technologies for discovering complex patterns in high volumes of data from IT systems, mainly data on systems' availability and performance, are called *IT operation analytics* (ITOA) technologies [1].

IT operation analytics technologies are used by IT operation teams for complex data analysis tasks including root cause analysis, service performance control, or service's impact analysis. Despite the advanced technologies used in ITOA domain, the data analysis still requires a significant portion of human interaction that slows down the analysis process, is a potential source of errors, and increases the financial cost of the analysis. Minimization of the amount of required human-machine interaction during the analysis process has been recognized as a challenge by Gartner in algorithmic ITOA domain [2].

The human interaction in ITOA tasks includes the formulation of complicated, multilevel queries that reveal complex patterns in the data. The formulation of the queries is a time demanding task, as an operator needs to define queries over various data sources, use different query languages, and merge and interpret the query results. Optimization of the querying approach in the ITOA domain would lead to the desired reduction of the volume of the necessary human interaction. *Complex event processing* (CEP) systems and *event processing languages* (EPL) have been successfully applied in other domains, such as commercial markets, to effectively analyze and query the complex data [3]. Application of the CEP in ITOA domain and definition of EPL language suitable for ITOA use-cases is expected to improve the analysis processes in ITOA.

In this paper, we demonstrate the application of the complex event processing approach in the IT operation analytics domain. We adapt existing T-Rex complex event processing system [4] for ITOA tasks. We propose several improvements for associated TESLA event processing language [5] to provide the sufficient expressiveness in network and application monitoring domains. The whole concept is demonstrated on root cause analysis use-cases derived from ITOA day-to-day practitioner's experiences.

II. Complex Event Processing Architecture

The general architecture of the T-Rex complex event processor is depicted in Figure 1. The event processor is built around the TESLA complex event specification language. The event processor adopts the client-server architecture, with a well defined binary protocol for exchanging events and event description between the server that does the event processing and its clients.

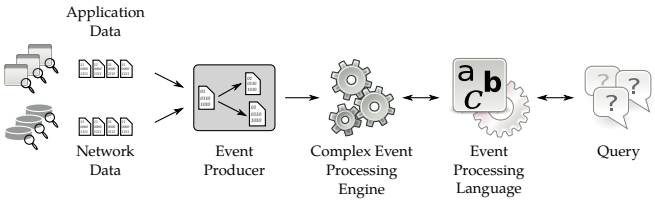

Fig. 1: Complex event processing architecture.

A thorough description of the TESLA event processing language is beyond the scope of this paper, and we refer the reader to see [5] for details. In general, its advantages for ITOA domain stem mainly from its expressiveness, support for recursion, and ability to process heterogeneous events without the need for event structure specification.

978-3-903176-15-7 © 2019 IFIP

T-Rex, which implements the TESLA language and enables plug-in integration of complex event processing into the ITOA domain, provides a number of technical advantages. Compared to other CEP systems, it has a small footprint and infrastructure requirements, while maintaining comparable performance. This can be exploited to decentralize the event processing and remove potential processing bottlenecks. It is also fully open source, which enabled us to alleviate some shortcomings of the language in regards to the ITOA domain.

For more natural processing of complex events, we have made some TESLA language extensions. These include support for a natural IP address type, support for array and string operations, and extensions of supported event timers. All these extensions will be open sourced soon.

III. Root Cause Analysis Demonstration

We present a sample of use-cases which are being tested on and deployed in large-scale academic and industrial installations. These installations are equipped with a subset of these monitoring tools: flow monitoring with heuristic event generation, intrusion detection systems, application performance monitoring and time-series analyzers.

We demonstrate two basic scenarios in which the root cause is a malware infiltration, which manages to avoid direct IDS detection during the attack phase. For each use case, we present an infection vector, the associated observable symptoms, which dictate the complex event logic, and the complex event description in the TESLA language. For posterity and to better convey the logic behind the analysis, a number of event attributes and event boilerplate are either hidden or taken as self-evident.

Use case I: Successful infection followed by an attempt to further spread malware. *Infection vector:* Weak authentication on target machine. *Symptoms:* Dictionary attacks preceding the infection, optional increase in outgoing traffic from the target, optional communication with blacklisted servers, IDS events originating from the target. *Complex event definition:*

```
Define CompromisedMachine(ip: inet)
From DictionaryAttack(dst_ip => $target) and
     last 10 OutTrafficInc(src_ip = $target) within 7 day
          from DictionaryAttack and
     last 10 Blacklist(src_ip = $target) within 7 day
          from DictionaryAttack and
     each IDSEvent(src_ip = $target) within 7 day
          from DictionaryAttack
Where ip := $target;
```

The event definition ties together all occurrences of dictionary attack attempts at some target with subsequent symptomatic events within one week. Thanks to the T-Rex mechanism, the one-week sliding window is transparently held for each occurrence of dictionary attack. All the events which are part of the given event chain are returned when an event detected by an IDS with the infected machine as a source is identified.

Use case II: Infected machine causes service disruption. *Infection vector:* Weak authentication on a target machine. *Symptoms:* Application performance monitoring (APM) index decrease, communication with the infected machine. *Complex event definition:*

```
Define MalwareServiceDisruption(ip: inet, source: inet)
From APMIndexDecrease(ip => $target) and
     CompromisedMachine($target in [OutTrafficInc.dst_ips,
                                    IDSEvent.dst_ips])
Where ip := $target, source := CompromisedMachine.ip;
```

This use case demonstrates the easy construction of event hierarchy and tying of high-level events together. In this case, it correlates a decrease in performance metric with communication with compromised machine. One clear advantage is that this complex event already has the complete event chain of *CompromisedMachine* event, thus the chain of events leading to performance decrease can be easily audited.

The functionality presented above can already be emulated by a lot of CEP tools. However, our approach enables a unified approach to heterogeneous event sources and more natural description of complex events and handling of time windows. Given the recursive nature of TESLA language, it enables easy creation of event hierarchy, and thanks to its small footprint, it can be easily deployed in infrastructure.

IV. Conclusions

In this paper, we demonstrate an application of the complex event processing approach into the IT operation analytics domain. On day-to-day ITOA use-cases, we show the formulation of the complex analysis problems in the TESLA event processing language and T-Rex engine. Our approach enables a natural description and processing of patterns that would require a series of advanced queries otherwise. We believe that the presented approach demonstrates the efficient and straightforward description of advanced patterns leading to the reduction of necessary human interaction presented in ITOA.

Acknowledgments

This research was supported by the Technology Agency of the Czech Republic under No. TA04010062 "Research and Development of Advanced Analytics Tools for Security and Performance Analysis of Network Infrastructure, Applications and Services".

References

[1] P. Adams and M. Govekar, "Hype Cycle for IT Operations Management," 2013. [Online]. Available: https://www.gartner.com/doc/2556718/hype-cycle-it-operations-management

[2] Gartner Inc., "Gartner says Algorithmic IT Operations Drives Digital Business," 2017. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-04-11-gartner-says-algorithmic-it-operations-drives-digital-business

[3] D. Luckham, "The power of events: An introduction to complex event processing in distributed enterprise systems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5321 LNCS. Springer, Berlin, Heidelberg, oct 2008, p. 3.

[4] G. Cugola and A. Margara, "Complex event processing with T-REX," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1709–1728, aug 2012.

[5] ——, "TESLA: A Formally Defined Event Specification Language," in *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '10. New York, NY, USA: ACM, 2010, pp. 50–61. [Online]. Available: http://doi.acm.org/10.1145/1827418.1827427