Solutions for Monitoring and Anomaly Detection in Dynamic IT Infrastructure: Literature Review

Jānis Grabis[1], Jānis Kampars[1], Krišjānis Pinka[1], Guntis Mosāns[1], Ralfs Matisons[1] and Artjoms Vindbergs[2]
[1]*Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia*
[2]*TET, Dzirnavu 105, Riga, Latvia*

Keywords: Infrastructure Monitoring, Network Topology, Streaming, Graph Analytics.

Abstract: Modern information technology infrastructure is highly complex, and its monitoring requires integration of different monitoring tools and management systems. That is especially important if monitoring data is to be used for predictive maintenance purposes. This paper identifies methods and technologies suitable for analysis of the information technology infrastructure. They are identified by means of literature review. The research questions considered are: 1) What methods are applicable for analysing the virtualized IT infrastructure related data from a technological point of view? 2) What architectural patterns and group of tools are appropriate for infrastructure data processing and analysis? and 3) What tools according to the identified categories in RQ3 can be used for storing and analysing topology graphs and metrics describing virtualized infrastructure? The research finding will serve as an input for further research activities on architectural design of the integrated monitoring solution and development of machine learning model for predictive maintenance.

1 INTRODUCTION

Modern cloud-based information technology (IT) infrastructure provides a large variety of services and serves many users. A typical large IT infrastructure generates millions of events per day at rates of about 100 events per second (Harper & Tee, 2019) and averaged sized cloud has around 1000 tenants and 100,000 users) (Majumdar et al., 2019). Security and reliability concerns should be addressed on the massive scale. Errors and faults occur in the complex IT infrastructure. They result in the potential loss of service to customers or cause extra work to maintain the infrastructure. Monitoring tools continuously track events in IT infrastructure and report the current and historical situation. They cover such aspects as performance including response time, availability, and uptime as well as security and other operating measures. Measures provided by component vendors as well as sniffing, scanning and benchmarking tools are using for monitoring purposes.

Open research issues in the monitoring of software-defined networks are support of adaptive measurements, real-time analytics, cyber-security support, cloud application integration, and quality-of-experience monitoring (Tsai et al., 2018). The taxonomy of cloud monitoring solutions describes various aspects of cloud computing monitoring many of which are concerned with cloud infrastructure management (Syed et al., 2017).

In the case of multi-functional data centres providing a variety of services, new holistic approaches to monitoring are needed (Natu et al., 2016). Monitoring context and data should be gathered from various systems and integrated to support the analysis of processes in the data centres. The performance monitoring and analysis system itself should possess the same degree of flexibility and adaptability as the virtualized and servitized infrastructure. Predictive maintenance and machine learning provide efficient means for managing security and reliability concerns though they have been rarely applied in the area of IT infrastructure (Su & Huang, 2018). Large scale applications of machine learning have been studied by Pacheco et al. (2019) in the context of network traffic classification.

The objective of the paper is to review existing methods and technologies for dynamic analysis of evolving IT infrastructure. The review reveals the current research gaps and suggests methods and technologies suitable for further development. A

Grabis, J., Kampars, J., Pinka, K., Mosāns, G., Matisons, R. and Vindbergs, A.
Solutions for Monitoring and Anomaly Detection in Dynamic IT Infrastructure: Literature Review.
DOI: 10.5220/0010446102240231
In *Proceedings of the 11th International Conference on Cloud Computing and Services Science (CLOSER 2021)*, pages 224-231
ISBN: 978-989-758-510-4

systematic literature study is conducted to achieve the objective. The research questions are motivated by the preliminary analysis of the infrastructure monitoring case, specifically dealing with the management and monitoring of virtualized storage infrastructure. This use case is investigated in collaboration with an industry partner.

The paper has five sections. Section 2 describes the motivational case. Section 3 establishes the research method. The detailed literature review is in Section 4. The findings of the literature review are summarized in Section 5. Section 6 concludes.

2 BACKGROUND

The existing monitoring systems and diagnostic tools target specific products and do not analyse the infrastructure as a whole, therefore, it is difficult to detect coherence among occurring incidents in different data centre sub-systems. The reactive IT infrastructure monitoring is available to limited extent and an overall analysis of the entire IT infrastructure with built-in predictive capabilities is not available.

A data centre analysed in the case study is complex and consists of physical and virtual components at multiple levels of abstractions, each with its own dynamically evolving topology. Physical infrastructure components such as computing nodes, power supply units, routers, switches, firewalls need to be considered. The virtual environment consists of software defined networks, software defined storage, hypervisor, virtual machines and containers. The topology of the data centre physical part is mostly static and evolves slowly, but the topology of the virtual environment changes dynamically.

An automatic and unified infrastructure monitoring system (Figure 1) that would provide automated incident root cause analysis and reduce the manual work is envisioned. This system observes the status of physical and virtual resources, records measurements, detects topology changes, provides incident root cause analysis and predictive maintenance. Data from monitoring tools can be divided into two groups – component related metrics and topologies. Both data sets need to be ingested into the data analysis component, which merges topologies on different levels of abstraction with the component related metrics. The result is a data centre level graph containing the corresponding metrics for each of the components and information about relations among the components. A database is used to store graphs, metrics and perform aggregations over historical values. The envisioned system provides stream processing capabilities for near-real time incident analysis. The results from the batch and stream processing can be observed in the monitoring user interface by the administrators of the data centre.

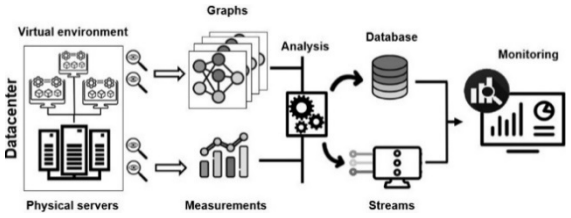

Figure 1: A proposal of integrated monitoring system.

3 METHOD

In order to identify methods and tools to implement the envisioned systems, a systematic review of the literature is performed. The literature review was done following the principles of a structured literature review (Kofod-Petersen, 2014). The process includes three stages – planning, conducting, and analysis of the literature reviews. To identify suitable sources, the choice of the literature used was determined by its: 1) relevance of the article, 2) the information contained in the citation, 3) the date of publishing. The main goal of this review is to gather information about the current situation in infrastructure monitoring, which is driven by the analysis of evolving topology infrastructure metrics. Research questions are specified to help achieve the goal of this research, which are shown in the Table 1.

Table 1: Overview of research questions.

RQ1	What methods are applicable for analysing the virtualized IT infrastructure related topologies and metrics?
RQ2	What architectural patterns are appropriate for the design of a horizontally scalable platform capable of analysing infrastructure topologies and metrics?
RQ3	What types of tools are appropriate for infrastructure data processing and analysis?

3.1 RQ1: Methods

Within the scope of this research two types of infrastructure related data are considered: 1) topology graphs – contains the topology of physical and virtual infrastructure components (e.g., disk arrays, software defined networks, virtual machines), and 2) numeric metrics – contains measurable properties describing

the status of a certain infrastructure component. Existing research papers mostly concentrate on one of the aspects – either topology or numeric metrics-based infrastructure analysis. A small group of researchers try to address both challenges within a single solution.

3.1.1 Topology Driven Infrastructure Analysis

The complexity of a modern cloud infrastructure topology is emphasized in a number of scientific articles. An IT infrastructure graph containing four levels of abstraction – application layer, container layer, virtual machine layer and hardware layer is mentioned in (Podolskiy et al., 2017). The authors propose an architecture that is aimed at dynamically adapting cloud application deployments on different infrastructure levels to meet quality of service (QoS) requirements. Dedicated tuning agents are used for each of the abstraction layers.

Each of the mentioned layers of abstraction can have its own graph with a significant level of complexity. Niwa et al., 2017 present a framework for identifying anomalies in software services of OpenStack cloud computing platform. Graph generator and statistics monitoring agents are installed and executed in each server. The framework is mostly implemented in Python, while Neo4J is used to store the topology graph data. The anomaly detection component uses K-means and MeanShift clustering algorithms.

Root cause analysis of an IT infrastructure failure is also addressed by Schoenfisch et al. (2018), who propose a Markov Logic Networks and abductive reasoning based solution. With the support of background knowledge expressed as ontologies the system enables users without any specific knowledge of a concrete infrastructure to gain viable insights in the case of an observed IT infrastructure incident. The proposed approach was implemented in RoCA, a tool providing a graphical user interface for modelling the infrastructure and conducting the root cause analysis.

Majumdar et al. (2019) audit cloud-based IT infrastructure for security purposes and propose a solution that is able to identify topology inconsistencies that might occur between multiple control layers in the cloud. The system gathers data from cloud management systems, cloud infrastructure system, data centre infrastructure components. The data collection is performed in batch mode. Authors use a Constraint Satisfaction Problem solver, namely Sugar (Tamura & Banbara, 2008) for validating the compliance of the cloud infrastructure.

The security threats caused by cloud platform misconfiguration or insider attacks are addressed by Bleikertz et al. (2015). The authors establish a security system, called Weatherman, which proactively analyses the intended cloud infrastructure configuration changes and risks associated with them and then either approves or rejects them, thus enforcing a variety of security and operational policies during run-time. The graph is constantly updated whenever changes in infrastructure configuration occur according to the approach presented in Bleikertz et al. (2014). The authors also define a threat model to identify possible misconfigurations and vulnerabilities in the graph and identify 95 methods which modified the topology or configuration in a way that can cause potential security threats.

A method to construct cloud-based IT infrastructure connectivity graph is presented by Mensah et al. (2017). The proposed solution retrieves the infrastructure topology and connectivity information in real-time from Cloud Management System and Software Defined Network controller. Logs from both systems are scanned to detect events that alter the infrastructure topology graph. The proposed system is validated by using OpenStack cloud computing platform.

3.1.2 Numeric Metrics-driven Infrastructure Analysis

A fault localization method for detecting failures of individual infrastructure elements can be based on the received operational status data and alerts (Harper & Tee, 2019). The authors state that a typical enterprise IT infrastructure might generate around 100 events per second. The work concentrates on the type of alerts that cause cascading errors in related infrastructure components. The cascading errors are detected without any knowledge of the infrastructure topology. In order to correlate groups of alerts and determine the root cause of cascading errors optimization techniques are employed for establishing a temporal similarity graph for the alerts.

MAYOR is a solution for processing communication system alarms (Mijumbi et al., 2019). The proposed system contains machine learning models for determining the persistence time of the alert dynamically and allowed to reduce the persistence time by 80% for 20% of all previously statically set alarm persistence times. The system was built using Apache Kafka, MongoDB and python data science tools such as sklearn, pandas, numpy. Another paper presents a mechanism for collecting

virtualized service communication metrics in a service agnostic way, which are then usable for detecting anomalies (Wallschläger et al., 2017). The proposed method, namely Deep Packet Inspection, concentrates on analysing communication protocol metrics. Anomaly detection and root cause analysis is also addressed by Lin et al. (2016). The paper proposes a method for virtualized cloud data centres. To address the scalability challenges the authors use Apache Spark. The anomaly detection is based on K-means clustering due to its simplicity and support for scalability requirements. The anomaly detection and root cause analysis functions are implemented in a Apache Spark based system, namely MonArch (Lin et al., 2015).

Another clustering-based anomaly detection solution is proposed by Cucinotta et al. (2020). The authors perform analysis of system-level metrics, mostly related to resource consumption patterns of virtual machines by using self-organizing maps (SOM) based approach. Miyazawa et al. (2015) use SOMs for supervised learning-based anomaly detection in IT infrastructure. Analysis of infrastructure can be used for cybersecurity threat identification (Farooq & Otaibi, 2018). Applicable machine learning algorithms are categorized as: data rate analytics; anomaly detection in process executions; and predicting user behaviour.

3.1.3 Topology and Numeric Metric-driven Infrastructure Analysis

A modular platform is able to process numeric metrics originating from entities such as infrastructure components while considering the topological relationships between the mentioned entities (Kampars & Grabis, 2018). Authors reference the original metrics as measurable properties and next level aggregates as context elements. Context elements can then be used to trigger various infrastructure related adjustments such as scaling virtual machines or triggering data replication. The topology information is entered manually in a web-based user interface as opposed to reading it from originating systems such as hypervisors as shown in other typology-driven infrastructure analysis research. Only basic aggregations such as average or max values withing a chosen time window are supported and more advanced use cases relying on machine learning are not reviewed as part of the paper. The system is based on Apache Kafka, Apache Spark and Apache Cassandra. Topology related information is stored in Cassandra together with numeric metrics and necessary metadata to generate Apache Spark jobs.

3.2 RQ2: Patterns

The event-driven architecture (Richards & Ford, 2020) is based on asynchronous communication and its main components are event processing nodes and queues or topics in which the results of event processing are written. Broker and mediator topologies are used for designing systems which are based on the event-driven architecture pattern. The advantage of this approach is that architecture does not have a central component for storing the business logic. The disadvantage is that it is difficult to implement a complex event orchestration logic, as it is scattered across services. In the mediator topology there is a central node ensuring implementation of complex orchestration logic what also makes the mediator a single point of failure and complicates horizontal scalability of the system.

Lambda (Persico et al., 2018) is a software architecture pattern associated with horizontally scalable big data processing platforms. A serving layer is responsible for indexing batch views, allowing to query them in a low-latency ad-hoc way. A single query can be run against results from both batch and real-time view. Big data platforms like Apache Spark and Apache Flink provide both batch and real-time data processing capabilities and the need for separate speed and batch layers was questioned. As a result, the Kappa architecture (Kreps, 2014) was proposed, containing only the real-time layer and no dedicated batch-processing layer. The architecture also employs a serving layer with querying capabilities for the streaming layer.

An experimental review of both competing architectural patterns was done by Sanla & Numnonda (2019). It was concluded that due to the dedicated batch layer Lambda contributes to higher resource consumption and cost. The advantages of Lambda are better resistance to data changes and reliability.

3.3 RQ3: Tools

This section reviews the following potentially useful tool groups and their corresponding characteristics - general-purpose stream processing platforms, message brokers, graph databases, graph streaming frameworks, time-series databases.

3.3.1 Stream Processing Platforms

Stream processing platforms are systems which ope-

rate on continuous data (Kambatla et al., 2014). In regard to IT infrastructure analysis examples of continuous data are metrics gathered from IT infrastructure components. The stream is not persisted before executing a query or data aggregation (Clemente & Lozano-Tello, 2018). Stream processing platform queries are run constantly based on a time interval or the amount of newly received data tuples (Hesse & Lorenz, 2016). Windowing algorithms are used to determine which data tuples should be processed together as part of a single query run (Lal & Suman, 2020). The stream process is made horizontally scalable by describing it as a directed acyclic graph, where separate processing tasks can be delegated to worker nodes in a stream processing cluster. An equal load distribution between worker nodes is an open research challenge (Nasir et al., 2015). Another issue is related to the order and time at which the data tuples arrive at the stream processing platform, which cannot be controlled by the stream processing platform (Hesse & Lorenz, 2016). Stream processing platforms address this challenge by introducing watermarks. Determining the right watermark time is an open research challenge (Onishi et al., 2020).

General-purpose stream processing platforms are potentially useful for infrastructure analysis due to their ability to process real-time metrics and late arriving data. The platforms also have limited graph processing capabilities.

3.3.2 Graph Databases

Graph processing is widely used in various areas of computer science such as machine learning, computational sciences, medical applications, social network analysis and corresponding graphs can contain up to several trillions of edges (Besta, Peter, et al., 2019). Although topology data can also be stored in regular databases this would lead to lost query optimization potential. Graph databases are superior since they support complex and rich graph models like Labelled Property Graph (Angles et al., 2017). Some of the graph databases also provide support for ACID compliant transactions (Besta, Peter, et al., 2019; Malewicz et al., 2010). Unfortunately, there is no common query standard, and a wide range of different query languages exist - SPARQL (Pérez et al., 2006), Gremlin (Rodriguez, 2015), Cypher (Francis et al., 2018). Two types of queries are usually supported – pattern matching queries (Zámečníková & Kreslíková, 2016) and business intelligence queries (Szárnyas et al., 2018). A subset of graph databases that could be particularly interesting for infrastructure analysis are platforms which support temporal or time-evolving graph analysis (Hartmann et al., 2017; Then et al., 2017; Vora et al., 2016).

Graph databases can be used for infrastructure topology analysis. Horizontal scalability and ability to store time-evolving graphs should be considered while choosing the appropriate system.

3.3.3 Graph Streaming Frameworks

The graph streaming frameworks perform processing of temporal, time-evolving, online, and dynamic graphs (Besta, Fischer, et al., 2019). While general-purpose stream processing platforms are concerned with processing continuous flow of data and have basic graph processing capabilities, graph streaming frameworks are tailored for near real-time analysis of dynamically evolving graphs where changes arrive in form of a continuous data flow. These tools differ from traditional graph processing platforms and databases like Pregel (Malewicz et al., 2010) and GraphX (Xin et al., 2014) which mostly consider static graphs. The graph streaming frameworks can be seen as hybrids between general purpose stream processing platforms and graph databases. Such systems need to deal with unique challenges like effective modelling and storage of dynamic datasets, efficient ingestion of a stream of graph updates in parallel with continuous graph queries (Besta, Fischer, et al., 2019). These systems usually do not track the historical state of the graph and concentrate on enabling low-latency graph updates and queries.

Since IT infrastructure topology is a dynamically evolving graph and anomalies should ideally be detected in near-real time, graph streaming frameworks could be beneficial for IT infrastructure analysis scenarios.

3.3.4 Time Series Databases

Time series are a finite or unbounded sequences of data points in increasing order by time (Jensen et al., 2017). Although traditional relational databases can be used to store time series data and various SQL extensions have been proposed for this purpose, they are not able to cope with high velocity and volume of data originating from sensor networks or large data centre infrastructure monitoring systems (Palpanas, 2016). NoSQL databases provide a solution which is based on better horizontal scalability, weakened relations and consistencies (Grolinger et al., 2013). As a result, a new generation of NoSQL-based databases which are particularly optimized towards storing and analysing large amounts of temporal data

have been proposed. These systems provide improved scalability to store large amounts of rapidly ingested time series data together with a capability of performing SCAN queries (Dunning & Ellen Friedman, 2014). It is hard to establish boundaries between NoSQL database management systems and time-series databases since both can provide similar functionality (Bader et al., 2017).

4 ANALYSIS

Scientific papers addressing IT infrastructure analysis were reviewed as part of RQ1 and led to a number of conclusions. There is almost no research done on combining topology analysis with analysing IT infrastructure characterizing metrics, although it is apparent that awareness of the topology would improve the accuracy of root cause analysis and identifying cascading events. Virtualization has introduced several new challenges for IT infrastructure analysis. Multiple layers of abstraction have greatly increased the number of entities that need to be monitored. The IT infrastructure components on different levels of abstraction might also have different ownership, which complicates the monitoring. Due to the large number of components and differential ownership, monitoring agents should be installed in a non-intrusive way and with little computing overhead. The monitoring data analysis platform needs to be highly scalable. The subject of IT infrastructure analysis for large cloud-based environments is not studied enough, since a significant amount of the existing researches propose solutions which are not appropriate due to scalability. Unsupervised and semi-supervised machine learning algorithms are more appropriate for cloud-based IT infrastructure analysis scenarios.

It was concluded that event-driven architecture patterns can be used in both Kappa and Lambda architecture. Kappa architecture is more lightweight and could provide lower latency and resource consumption. Lambda is more appropriate for use cases where complex machine learning algorithms and data pre-processing need to be used – such as in IT infrastructure analysis.

The review of the appropriate tool groups has led to the following conclusions: 1) to select a message broker, the priorities for latency, throughput, reliability and horizontal scalability as well as the supported messaging models need to be defined; 2) Watermarks are used to address the issue of late-arriving data tuples in stream processing platforms, however determining the right watermark size is challenging; 3) Although stream processing platforms have some graph processing capabilities, more specialized graph streaming frameworks can be used to analyse dynamically evolving graphs; 4) Traditional graph databases provide advanced means for graph-based analytics; however, they fail to support low latency queries for dynamically evolving graphs; and 5) time series querying functionality varies greatly among the tools, therefore, the set of required type of queries for analytic purposes needs to be identified prior to choosing the most appropriate database management system.

5 CONCLUSION

This literature review has identified the current research gaps and state-of-the-art in IT infrastructure analysis. The findings of the review will be used to design an integrated platform for IT infrastructure monitoring and predictive maintenance. Concrete tools belonging to the identified groups of tools are to be tested and selected for their inclusion in the solution. That will serve as an input for architecture development and implementation of the monitoring and prediction platform.

ACKNOWLEDGEMENT

This research is funded by European Regional Development Fund Project Nr. 1.1.1.1/19/A/003 “Development of integrated monitoring and predictive maintenance solution for dynamically evolving IT infrastructure” Specific Objective 1.1.1 “Improve research and innovation capacity and the ability of Latvian research institutions to attract external funding, by investing in human capital and infrastructure” 1.1.1.1. measure “Support for applied research” (round No.3).

REFERENCES

Angles, R., Arenas, M., Hogan, A., & Reutter, J. (2017). Foundations of Modern Query Languages for Graph. 50(5).

Bader, A., Kopp, O., & Falkenthal, M. (2017). Survey and comparison of open source time series databases. Lecture Notes in Informatics (LNI), *Proceedings - Series of the Gesellschaft Fur Informatik (GI)*.

Besta, M., Fischer, M., Kalavri, V., Kapralov, M., & Hoefler, T. (2019). Practice of streaming and dynamic graphs: Concepts, models, systems, and parallelism.

ArXiv, 1–16.

Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2019). Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *CoRR*, abs/1910.0.

Bleikertz, S., Vogel, C., & Groß, T. (2014). Cloud radar: Near real-time detection of security failures in dynamic virtualized infrastructures. *ACM International Conference Proceeding Series*, 2014, 26–35.

Bleikertz, S., Vogel, C., Gross, T., & Mödersheim, S. (2015). Proactive security analysis of changes in virtualized infrastructures. *ACM International Conference Proceeding Series*, 7-11 Dec, 51–60.

Clemente, P. J., & Lozano-Tello, A. (2018). Model Driven Development Applied to Complex Event Processing for Near Real-Time Open Data. *Sensors*, 18(12).

Cucinotta, T., Lanciano, G., Ritacco, A., Vannucci, M., Artale, A., Barata, J., Sposato, E., & Basili, L. (2020). Behavioral analysis for virtualized network functions: A som-based approach. *CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science*, Closer, 150–160.

Dunning, T., & Ellen Friedman. (2014). Time Series Databases: New Ways to Store and Access Data. *O'Reilly Media, Inc*.

Farooq, H. M., & Otaibi, N. M. (2018). Optimal machine learning algorithms for cyber threat detection. *Proceedings - 2018 UKSim-AMSS 20th International Conference on Modelling and Simulation*, UKSim 2018, 32–37.

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Selmer, P., & Taylor, A. (2018.). Cypher : An Evolving Query Language for Property Graphs. SIGMOD '18: *Proceedings of the 2018 International Conference on Management of Data*, 1433–1445.

Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. M. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing*, 2(1).

Harper, R., & Tee, P. (2019). A method for temporal event correlation. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2019*, 13–18.

Hartmann, T., Fouquet, F., Jimenez, M., Rouvoy, R., & Le Traon, Y. (2017). Analyzing Complex Data in Motion at Scale with Temporal Graphs. The *29th International Conference on Software Engineering & Knowledge Engineering (SEKE'17)*, Jul 2017, Pittsburgh, US, pp. 6.

Hesse, G., & Lorenz, M. (2016). Conceptual survey on data stream processing systems. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, 2016, 797–802.

Jensen, S. K., Pedersen, T. B., & Thomsen, C. (2017). Time Series Management Systems: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(11), 2581–2600. 2.

Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics . *Journal of Parallel and Distributed Computing*, 74(7), 2561–2573

Kampars, J., & Grabis, J. (2018). Near Real-time Big-data Processing for Data Driven Applications. *Proceedings - 2017 International Conference on Big Data Innovations and Applications, Innovate-Data 2017*, 35–42.

Kofod-petersen, A. (2014). How to do a structured literature review in computer science. *Researchgate*, May 2015, 1–7.

Kreps, J. (2014). Questioning the Lambda Architecture. The Lambda Architecture has its merits, but alternatives are worth exploring. *O'Reilly Media on line*, July.

Lal, D. K., & Suman, U. (2020). A Survey of Real-Time Big Data Processing Algorithms. *Lecture Notes in Mechanical Engineering*, 3–10.

Lin, J., Ravichandiran, R., Bannazadeh, H., & Leon-Garcia, A. (2015). Monitoring and measurement in software-defined infrastructure. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, 742–745.

Lin, J., Zhang, Q., Bannazadeh, H., & Leon-Garcia, A. (2016). Automated anomaly detection and root cause analysis in virtualized cloud infrastructures. *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Noms*, 550–556.

Majumdar, S., Madi, T., Wang, Y., Tabiban, A., Oqaily, M., Alimohammadifar, A., Jarraya, Y., Pourzandi, M., Wang, L., & Debbabi, M. (2019). Cloud security auditing. In *Advances in Information Security* (Vol. 76).

Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010). Pregel : A System for Large-Scale Graph Processing. 135–145.

Mensah, P., Dubus, S., Kanoun, W., Morin, C., Piolle, G., & Totel, E. (2017). Connectivity graph reconstruction for networking cloud infrastructures. *2017 IEEE 16th International Symposium on Network Computing and Applications*, 1–9.

Mijumbi, R., Asthana, A., Bernal, C., & Castejon, M. (2019). MAYOR: machine learning and analytics for automated operations and recovery. *Proceedings - International Conference on Computer Communications and Networks, ICCCN, 2019-July*.

Miyazawa, M., Hayashi, M., & Stadler, R. (2015). VNMF: Distributed fault detection using clustering approach for network function virtualization. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, 640–645.

Nasir, M. A. U., De Francisci Morales, G., García-Soriano, D., Kourtellis, N., & Serafini, M. (2015). The power of both choices: Practical load balancing for distributed stream processing engines. *Proceedings - International Conference on Data Engineering, 2015*-May, 137–148.

Natu, M., Ghosh, R. K., Shyamsundar, R. K., & Ranjan, R. (2016). Holistic Performance Monitoring of Hybrid Clouds: Complexities and Future Directions. *IEEE Cloud Computing*, 3(1), 72–81.

Niwa, T., Kasuya, Y., & Kitahara, T. (2017). Anomaly detection for openstack services with process-related

topological analysis. *2017 13th International Conference on Network and Service Management, CNSM 2017*, 1–5.

Onishi, T., Michaelis, J., & Kanemasa, Y. (2020). Recovery-conscious adaptive watermark generation for time-order event stream processing. *Proceedings - 5th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2020*, 66–78.

Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., & Aguilar, J. (2019). Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey. *IEEE Communications Surveys and Tutorials*, 21(2), 1988–2014. 7

Palpanas, T. (2016). Big Sequence Management: A glimpse of the Past, the Present, and the Future. In R. M. Freivalds, G. Engels, & B. Catania (Eds.), *SOFSEM 2016: Theory and Practice of Computer Science* (pp. 63–80). Springer, Berlin.

Pérez, J., Arenas, M., & Gutierrez, C. (2006). Semantics and Complexity of SPARQL. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, & L. M. Aroyo (Eds.), *The Semantic Web - ISWC 2006* (pp. 30–43). Springer, Berlin.

Persico, V., Pescapé, A., Picariello, A., & Sperlí, G. (2018). Benchmarking big data architectures for social networks data processing using public cloud platforms. *Future Generation Computer Systems*, 89, 98–109.

Podolskiy, V., Gerndt, H. M., & Benedict, S. (2017). QoS-based cloud application management: Approach and architecture. *CrossCloud 2017 - 4th Workshop on CrossCloud Infrastructures and Platforms, Colocated with EuroSys 2017*.

Richards, M., & Ford, N. (2020). Fundamentals of Software Architecture. *O'Reilly Media, Inc*.

Rodriguez, M. A. (2015). The Gremlin Graph Traversal Machine and Language.

Sanla, A., & Numnonda, T. (2019). A comparative performance of real-time big data analytic architectures. *ICEIEC 2019 - Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication*, 674–678.

Schoenfisch, J., Meilicke, C., Stülpnagel, J. von, Ortmann, J., & Stuckenschmidt, H. (2018). Root cause analysis in IT infrastructures using ontologies and abduction in Markov Logic Networks. *Information Systems*, 74, 103–116.

Su, C. J., & Huang, S. F. (2018). Real-time big data analytics for hard disk drive predictive maintenance. *Computers and Electrical Engineering,* 71, 93–101.

Syed, H. J., Gani, A., Ahmad, R. W., Khan, M. K., & Ahmed, A. I. A. (2017). Cloud monitoring: A review, taxonomy, and open research issues. In *Journal of Network and Computer Applications* (Vol. 98, pp. 11–26). Academic Press.

Szárnyas, G., Prat-pérez, A., Averbuch, A., Marton, J., Paradies, M., Kaufmann, M., et al. (2018). An early look at the LDBC Social Network Benchmark' s Business Intelligence workload. *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems and Network Data Analytics*, No. 9, 1–11

Tamura, N., & Banbara, M. (2008). {Sugar}: A {CSP} to {SAT} Translator Based on Order Encoding, 65–69.

Then, M., Stephan, G., Neumann, T., & Kemper, A. (2017). Automatic Algorithm Transformation for Efficient Multi-Snapshot Analytics on Temporal Graphs. 10(8), 877–888.

Tsai, P. W., Tsai, C. W., Hsu, C. W., & Yang, C. S. (2018). Network Monitoring in Software-Defined Networking: A Review. *IEEE Systems Journal*, 12(4), 3958–3969.

Vora, K., Gupta, R., & Xu, G. (2016). Synergistic Analysis of Evolving Graphs. ACM Trans. Archit. Code Optim., 13(4).

Wallschläger, M., Gulenko, A., Schmidt, F., Kao, O., & Liu, F. (2017). Automated Anomaly Detection in Virtualized Services Using Deep Packet Inspection. *Procedia Computer Science*, 110, 510–515.

Xin, R. S., Crankshaw, D., Dave, A., Gonzalez, J. E., Franklin, M. J., & Stoica, I. (2014). GraphX: Unifying Data-Parallel and Graph-Parallel Analytics. http://arxiv.org/abs/1402.2394

Zámečníková, E., & Kreslíková, J. (2016). Comparison of platforms for high frequency data processing. *2015 IEEE 13th International Scientific Conference on Informatics, INFORMATICS 2015 - Proceedings*, 296–301.