Automated processing of monitoring data for proactive root cause analysis in service-based systems

Elias Detrois
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Stuttgart, Germany
st173930@stud.uni-stuttgart.de

Matthias Weiß
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Stuttgart, Germany
matthias.weiss@ias.uni-stuttgart.de

Michael Weyrich
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Stuttgart, Germany
michael.weyrich@ias.uni-stuttgart.de

***Abstract*—The transition from monolithic software to microservices promises advantages in scalability and continuous development, accompanied by the challenge of correctly understanding and efficiently addressing errors in distributed, service-based systems. Incidents in these systems can incur substantial costs, necessitating rapid root cause analysis (RCA) to restore normal operations. Conventional RCA methods, while offering assistance, demand significant manual effort and prove challenging as well as time-consuming even for experienced engineers. In response to recent advances in the area of machine learning, this paper explores novel approaches to enhance the speed and interpretability of RCA in service-based systems. Based on a structured literature review, current trends and research gaps are extracted. In addition, a concept is developed to improve the real-time behavior and comprehensibility of RCA results. The feasibility of the concept is evaluated on the basis of a value proposition and stakeholder analysis. By integrating generative AI to generate root cause explanations and propose mitigating actions, this study seeks to advance the interpretability of RCA results. In combination with improvements to RCA efficiency and real-time processing this reduces the cost and complexity associated with service-based systems.**

***Keywords*—Root Cause Analysis, service-based, real-time, Artificial Intelligence, cloud solutions, interpretability**

I. Introduction

Classic monolithic software architectures consist of closely linked processes executed as a single application. This results in an executable that is easy to deploy and test in one centralized place. However, in the event of load peaks, the entire application must be scaled accordingly to handle extra traffic, even if the load is only concentrated on one process within the system. Moreover, the growing complexity of such monolithic code bases makes it increasingly difficult to implement modifications and keep track of the code. Additionally, maintenance and updates to the functionality always require new releases of the entire application. [1]

In order to meet the increasing demand for connectivity, scalability and continuous development, modern cloud solutions rely on microservices architecture. This involves dividing a complex system into individual systems that are as small and independent as possible. System functions are realized through communication between the required services. This separation makes it easier to scale individual services, develop them further or even replace them during operation. [2]

However, the advantages of microservices architectures also bring new challenges. Due to the huge number of services involved and the continuous development of individual components, errors can occur. These errors are then difficult to localize and understand so suitable repair measures can be initiated. [3] Incidents in large service-based cloud systems can be very expensive, especially in the direct customer environment or if production comes to a standstill. [4] In these cases, service providers should be able to react quickly to restore normal operation. [5] Errors in the form of downtime can, according to the Ponemon Institute [6], amount to almost $9,000 per minute, which is why fast, efficient and precise methods for root cause analysis (RCA) offer great savings potential for companies.

Conventional methods for RCA already offer the possibility of simplifying the analysis steps. However, they still require significant time and effort for manual examinations of data sources such as logs and traces and are a challenge even for experienced engineers.[7]

New approaches to real-time processing of log and trace data offer the opportunity to accelerate RCA and greatly reduce response times for affected teams. In combination with recent advancements in machine learning, it is possible to move beyond reactive strategies towards predictive approaches that already recognize potential problems and warn of failures in a timely manner. For this reason, this thesis examines methods that improve the real-time capability of RCA and enable engineers to better understand and predict incidents.

The main objective of this research is to investigate such approaches that utilize artificial intelligence (AI) for real-time processing of log and trace data to improve RCA in dynamically evolving service-based systems. The integration of AI techniques is intended to improve the efficiency and accuracy of problem detection and resolution. It also aims to enable proactive troubleshooting by predicting incidents and ultimately increase the overall reliability and performance of these systems. In a further section, this research will examine the proposed methods in terms of technology transfer and economic aspects. The aim is to evaluate the value proposition of the technology and its potential for use in an industrial environment. This includes identifying the specific stakeholders and presenting the opportunities and risks for the technology in practice.

II. Methodology

A systematic literature review (SLR) is a method of comprehensively identifying, evaluating, and summarizing existing research relevant to a particular research question or topic. The aim of an SLR is to provide a comprehensive and neutral overview of a field of research and to record the current state of research as comprehensively as possible. It can also be used to identify gaps in research on a particular topic. Conducting an SLR consists of several steps, which are described in the following sections.

A. Defining a research question

The first step in conducting a systematic literature review is to define the research question and establish inclusion and exclusion criteria. These so-called "requirements" determine the basis on which research studies are classified as relevant for addressing the specified research question. Furthermore, exclusion criteria can also be defined. These criteria help to maintain the objectivity of the review and ensure that the selected studies are in line with the research question. Five requirements were defined for this study.

1) Real-time behavior

This requirement evaluates the selected timeslot used for assessment in RCA. Depending on the approach this can range from static slots to event-driven selection or even real-time updates to the system model preserving historical data.

2) Scope of the analysis

Depending on the system, it may be challenging to gain visibility into specific services. Therefore, it is essential to evaluate the granularity and level of detail provided by each approach.

3) Interpretability of the results

Also considered as an important aspect is the presentation and interpretability of analysis results. Improvements may include the generation of root cause explanations or suggestions for corrective actions.

Approaches should simplify RCA and enhance ease of use, reducing efforts for service personnel. Additionally, the integration of expert knowledge or feedback is considered as it can improve decision-making in the long run.

4) Intelligence and proactivity of the approaches

Proper implementation of modern AI methods can significantly enhance the results of RCA. Integrating such methods is expected and can improve modern tools' ability to detect patterns in massive amounts of data. This could eventually lead to a shift from reactive analysis to proactive and predictive strategies.

5) Adaptability

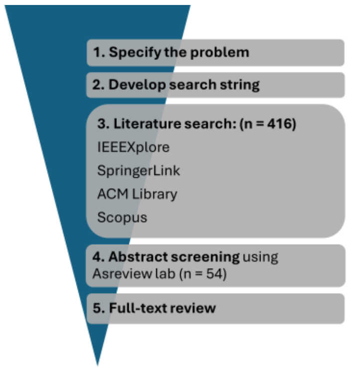
Lastly, adaptability is examined as a criterion for inclusion. Approaches that can be easily or automatically adapted can be used in multiple environments. These agile approaches minimize the need for custom-tailored solutions for each use case, ensuring their relevance and long-term impact.

B. Develop and validate the search strategy

Once the research question and the requirements have been defined, a systematic search strategy is developed to identify relevant literature in the databases. The search for this paper queried the electronic databases IEEE Xplore, SpringerLink, ACM Library, and Scopus using a specially developed search string. A small amount of gray literature was also included.

The search string consists of keywords, Boolean operators, and specific subject terms that effectively narrow down the search to a particular subject area. The query was revised and refined through several iterations until the desired scope of results was achieved. The terms of the query remained mostly the same between revisions, but the Boolean operators linking them differed slightly with each new version.

Fig. 1. SLR methodology



Final version of the search string: n=1040 hits with 416 hits between 2020-2023

("Root Cause Analysis" OR RCA) AND (causality mining OR "site reliability engineering" OR SRE OR observability) AND (service-based OR distributed) AND (systems OR infrastructure)

C. Screening for inclusion

All identified articles underwent a screening process to determine their eligibility for inclusion in the review. The screening process involved two stages: title and abstract screening, followed by full-text assessment. The software tool ASReview Lab was used to efficiently narrow down a collection of 416 papers to a more manageable subset of 54 for in-depth analysis. To utilize this tool the user inputs data into the adaptive learning algorithm of the tool by labeling the literature as “relevant” or “not relevant” based on their title and abstract. ASReview Lab then utilizes this data to prioritize the most applicable articles to feed to the user. Once no more relevant literature can be identified in the feed, the user can conclude the search. [8] ASReview Lab enabled efficient screening of the literature, resulting in a focused and relevant set of papers for full-text analysis.

D. Extract data

Data extraction is the final step in collecting relevant information from each included study. In this research, data extraction consisted of reading the most relevant papers and identifying research gaps while becoming more familiar with the topic. Additionally, the goal of this study was to create a hypothetical product to address the challenges of current approaches to RCA. For this reason, the extraction process was streamlined to find connections between research gaps and proposed solutions. This was later combined with a search for more specific literature needed for the use case and proposed concept.

III. RESEARCH-GAPS

As mentioned in the introduction service-based systems are abundant and used in all cloud environments to provide the expected quality of service. The importance of efficient methods and tools for anomaly detection and root cause analysis. This section outlines the identified research gaps related to RCA that were identified during the data extraction phase of this structured literature review.

A. Interpretability

Understanding the results of today's state-of-the-art tools is often a challenge, even for experienced personnel. Current software solutions for root cause analysis are often difficult to set up, and the results can be hard to interpret. [9, 10]

This means that a significant amount of knowledge and time is required to inspect an issue in depth, even after the responsible service has been identified. In a highly diverse tech stack, knowledge is spread across many teams. This creates a knowledge gap between the different analysts which can result in more delay gathering all the necessary information to understand an anomaly. [3]

One of the biggest challenges with current RCA approaches is presenting the data to the responsible expert. While many approaches can quickly provide context for observed anomalies, they lack significant guidance after the analysis.

B. Scope

The selection of the correct scope of analysis in the case of a failure is very important for effective RCA. The accuracy and relevance of the results depend on the inclusion of the defining data for an observed anomaly.

Due to the large number of services in such service-based environments, it is difficult to select the right set of data in the analysis that is representative of the current state of the system. At the same time, it is important to keep the data included into the analysis to a minimum to maintain a certain level of efficiency, especially when real-time behavior is desired. [11]

This careful selection process ensures that RCA remains targeted, comprehensive, and capable of providing actionable insights for troubleshooting and optimization.

C. Real-time analysis

Traditional methods, often rely on the analysis of static models or periodic evaluations of the system state. As a result, they struggle to keep up with the real-time demands of modern systems, leading to delays in problem identification and resolution. Moreover, the sheer volume of data generated in dynamic environments pose a significant hurdle, contributing to additional latency. [12]

Additionally, the integration of diverse data sources, such as logs, traces, and metrics, requires cohesive real-time analytics strategies to evaluate information from multiple channels effectively. The need to adapt to sudden spikes in traffic (scalability) further complicates the real-time analytics landscape. [10] Overcoming these challenges requires innovative approaches.

D. Proactive analysis

Not only do current approaches fail to detect and understand complex systems in real-time, as a result they also face limitations in proactively detecting potential failures before they manifest themselves inside the system. This is primarily due to conventional tools operating reactively, starting the analysis after an anomaly has been detected and analyzing historical metrics. This retrospective process inhibits their ability to foresee or even prevent potential failures beforehand.

Furthermore, the delay between data collection and analysis as described in the section above contributes to the failure to detect issues pre-emptively. The lack of adaptability to evolving environments and unforeseen circumstances further limits their predictive capabilities. The interdependencies, scale, and constant changes in system configurations make it challenging to develop predictive models that capture the full spectrum of potential failure scenarios.

IV. PROPOSED APPROACH

This section is intended to convey the proposed concept for a more intelligent and user-friendly RCA tool by combining ideas, concepts and implementations found across the different research papers. It is however important to note, that this is only a concept that was required for this research paper, but no proof-of-concept was implemented.

A. Event Handler

As a first step, all approaches must recognize anomalies in the system. Since the detection process is not considered in this paper, it is only necessary to determine how to act after an anomaly has been detected. In manual processes, these steps involve a great deal of experience and expertise on the part of the responsible experts. These experts always define their action steps depending on the observed anomaly. However, there are often internal working guidelines for predefined recommendations based on the error at hand. Chen *et al.* [7] try to integrate expert knowledge by imitating this decision-making process with "incident handlers". Although these incident handlers must be created manually, they offer a way of influencing the system and incorporating knowledge into the analysis, and they can also carry out simple and repetitive remedial measures quickly, efficiently and automatically. [7]

The proposed approach can incorporate these handlers to prepare all data for processing with LLMs. They can also reduce unnecessary computation time for recurring problems and submit the finalized prompts to an LLM.

B. Data fusion

In their research Yu *et al.* [10] incorporate an interesting technique in the RCA. They argue that the amount and heterogeneity of the system data contributes heavily to the duration and inaccuracy observed in many approaches. Many parts of this multimodal data cannot be used sufficiently by the approaches because they cannot process them. This is why the authors suggest pre-processing the data in one step. Instead of passing on the raw data for analysis, an "event" is created after an anomaly is detected. This event combines the heterogeneous data into a uniform format with uniform timestamps. This new representation allows for more information to be used in the analysis while also providing a standardized format. [10]

This method appears to be particularly advantageous for the subsequent combination with LLMs, which above all require good prompt engineering. This event representation therefore serves as the first preparation step in order to optimally prepare the data and extract as much information as possible.

C. Incremental learning

One of the components of RCA approaches is usually the creation of a digital representation of the underlying structure of a system. This is usually done with a graph representation where the vertices represent objects like nodes or events and the edges are used to model their relations to each other. This helps maintain a view of the system and allows for the use of graph theory to improve root cause discovery.

To deal with the issue of static models and time slot selection Wang *et al.* [12] propose incremental graph building. The process consists of three main steps.

1) Trigger point detection

2) Incremental graph learning

3) Root cause localization

This means that the proposed framework monitors the System KPIs nearly in real-time and once a change has been detected in the state or communication of the system the next step is triggered. In this step the model needs to distinguish between relations that have remained the same and the new relations introduced in the current batch of data. After this extraction of changes is finished, the information can be merged into the existing graph. This results in a continuously updated representation of the system, and it creates the possibility for online RCA and much faster root cause discovery times.

This approach brings improvements to current state-of-the-art methods by enabling online RCA in real-time. According to the authors it can improve efficiency dramatically by only updating based on changes in the system. Another advantage is the consideration of all available historic data instead of fully retraining and refitting a new model for every anomaly on a selected batch of data. [12]

Incorporating this strategy for the creation of real-time graph models would therefore increase efficiency and response time of the proposed concept.

D. Graph pruning

To expand on the incremental graph building described in the previous section and improve efficiency even further graph pruning could also be incorporated.

Ding *et al.* [13] describe an approach that utilizes reinforcement learning to learn a pruning policy. This policy can be used to effectively eliminate redundant information from the dependency graph. Not all services are affected or even relevant in the context of a failure and subsequent RCA. If this redundant information can be eliminated from the data before further processing, without affecting the analysis, it can save time and resources. Various characteristics of root causes and trace data are used to determine the relevance of components and data in the analysis step. They can be arranged in three groups. [13]

1. *Latency-based:*
   Different latency metrics are used to determine whether a service is affected.
2. *Anomaly-based:*
   Based on the number and type of anomalies only services with high anomaly indicators are considered further.
3. *Correlation-based:*
   Lastly correlation metrics are used to determine whether observed anomalies/ latencies are likely to be related.

By implementing these preprocessing steps into the proposed concept, the incremental graph building could be even more efficient and would only need to be triggered if a change is deemed relevant. Not only will it be more efficient but if implemented correctly, the relevant part of system could be analyzed in more detail without using too much computing resources.

E. Explanation generation

Now that the discussion of pre-processing is complete, it is time to introduce the core of the proposed method. As previously stated in the introduction, the approach will utilize modern AI methods to present the results more effectively and enhance their interpretability. For instance, LLMs have demonstrated significant potential in identifying patterns in vast amounts of data. This step in the concept is designed to process the prepared data and automatically generate explanations and suggestions for experts. Chen *et al.* propose a procedure that aims to automate the diagnostic process, relieving responsible engineers of diagnosis and enabling immediate reaction and mitigation.

The following procedure is suggested in connection with the methods explained in this chapter. Figure 2 shows an overview of the concept.

1. Incident handler queries and collects the necessary data using data fusion or responds with immediate action if possible.
2. Incremental Graph building is used to have an accurate representation of the system state and possibly of event causalities. [3]
3. The incident handler triggers an automatic summation of the text and traces collected by the handler. This reduces noise and results in better predictions. [7]
4. An event is created with all the information collected and the summary and the LLM is prompted to make a prediction for possible root causes with an explanation and mitigation suggestion. [7]

This procedure improves the readability of the results and makes it easier for less experienced personnel to understand the situation and take immediate action.

One aspect that still needs to be tackled in this context is the interaction of dependency graphs and LLMs. The graphs offer the possibility of incorporating external knowledge into the decision-making process of the LLM. However, this approach, as well as the generative AI methods, is still quite new and there is still little research in this area. However, novel approaches can already be found as to how this problem could be solved and how domain-specific knowledge can be introduced. [14, 15]

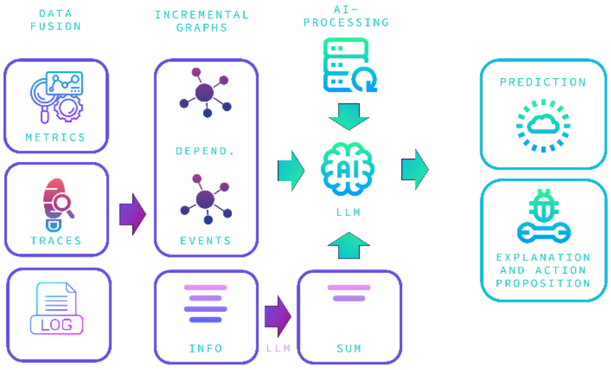


Fig. 2. Proposed concept overview

V. Value Proposition and Stakeholder Analysis

After Chapter 3 identified a research gap in the area of RCA, the following section focuses on offering a possible solution by utilizing recent advances in the area of real-time root cause discovery, generative AI, and efficiency optimizations. Therefore, this chapter first introduces the new approach XPRED and its potential stakeholders. Then, the information about both the approach and the stakeholders, especially the potential customers, is combined by a value proposition. The solution is validated by analyzing whether there is a problem-solution fit.

Checking the feasibility of new technologies and research results is a crucial step in the innovation process to ensure that new ideas can be implemented in practice. Feasibility analyses must include technical, economic, and operational considerations. This validation process reduces the risk of investing resources in concepts that are not technically feasible, financially viable or impractical. The aim of this step is to harmonize technological advances with market requirements in order to create successful solutions from novel concepts.

The proposed concept was developed from identified research gaps in the area of RCA for service-based systems. The focus is on improving interpretability, real-time analysis, and operational efficiency by integrating LLMs. Root cause analysis within service-based architectures generally faces the challenge of deciphering complicated system behavior and usually involves many manual steps. While existing solutions on the market advertise a wide range of functions and AI analyses, their initial setup is often complicated, and their use requires a great deal of expertise. Nevertheless, recognizing patterns and causalities often falls largely on the responsible engineers.

It is precisely this complexity that the proposed system addresses by incorporating advanced LLMs to make the analysis more understandable and the results more interpretable. These LLMs, such as ChatGPT, have already shown great potential in processing textual information. This can provide the user with targeted insights into the underlying data and recognize patterns in the case of recurring errors. In addition, the natural language capabilities of these models are used to generate comprehensible summaries of incidents and thus make the results of the analyses easier to understand.

Another advantage of the proposed solution lies in its ability to provide real-time insights, a crucial feature in the context of dynamically evolving systems. The continuous updating of the underlying graph makes it possible to capture this dynamic development of the system. On the other hand the graph does not have to be rebuilt for each incident which makes the analysis more efficient and avoids problems associated with the choice of a fixed time slot for viewing the system data.

By utilizing these technologies, the proposed approach facilitates the immediate identification of emerging issues and enables quick and informed decision making to mitigate potential disruptions. This real-time analytics capability is critical to minimizing downtime and optimizing the overall performance of service-based architectures. In addition, the inclusion of LLMs increases the accuracy and depth of root cause analysis. These language models not only improve interpretability, but also enable a better understanding of complex patterns and anomalies within the system. It is essential that the use of LLMs is in line with a scientific and data-driven approach. This ensures that the diagnostic process is underpinned by robust methods and empirical evidence.

To sum up, the proposed approach provides a comprehensive and efficient solution to the challenges associated with RCA in service-based systems. It goes beyond traditional methods and provides a platform that not only identifies problems but also systematically interprets them, thus improving the overall resilience and reliability of service-based architectures. This offers great opportunities in connection with a possible implementation of this System in the automotive sector. Particularly in the area of car-to-x communication and the associated cloud management, a comprehensive stakeholder analysis is essential in order to identify the various players and their interests in the successful introduction and use of this innovative technology.

First and foremost, car manufacturers are the most important players in this scenario. The integration of this software into their infrastructure gives manufacturers an advanced tool to quickly diagnose and solve problems related to car-to-x communication. In particular, the real-time analysis and interpretation capabilities of the software offer the automotive industry the opportunity to improve vehicle connectivity and prepare the communication infrastructure for new technologies such as autonomous driving and intelligent traffic management. The product is designed to make complex systems easier to maintain and generally more manageable. Making the operation of such systems more accessible can encourage further innovation by reducing companies' concerns about downtime.

In this environment, technology integrators and software developers in automotive groups are the key players. The proposed system is designed for interoperability and seamless integration and presents a valuable tool for developers who want to improve Car-to-X communication capabilities in their applications and maintain the necessary infrastructure. Early adopters who are testing and introducing new technologies in their companies must be addressed in particular.

The software is also a useful tool for automotive service providers and maintenance teams. Its advanced root cause analysis features enable quick and accurate diagnosis of communication faults, facilitating optimized maintenance procedures and minimizing vehicle downtime. This has a direct impact on operational efficiency and customer satisfaction.

The end users, the drivers and passengers of connected vehicles, are also an important stakeholder group. The software improves their overall experience by ensuring the reliability of car-to-x communication, which is essential for functions such as traffic alerts, navigation optimization and, above all, enhanced safety features. Improving the interpretability and real-time analysis of communication issues, contributes to a safer and more efficient driving experience, meeting the growing expectations of modern consumers for high connectivity and functionality.

In summary, the success of the proposed system in the automotive sector, especially in the context of Car-to-X communication, requires a nuanced understanding of the various stakeholders and their respective interests. Car manufacturers, regulators, service providers, end users and technology integrators can all benefit from the advancement of reliability of service-based systems. Making these complex and dynamically evolving systems more accessible for companies catalyzes advances in modern service-based cloud systems and contributes to the realization of a connected and intelligent ecosystems in many domains.

References

[1] Amazon Web Services, Inc. "Was sind Microservices? | AWS." Zugriff am: 20. Januar 2024. [Online.] Verfügbar: https://aws.amazon.com/de/microservices/

[2] L. Wang *et al.*, "Root Cause Analysis for Microservice Systems via Hierarchical Reinforcement Learning from Human Feedback," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach CA USA, A. Singh et al., Hg., 2023, S. 5116–5125, doi: 10.1145/3580305.3599934.

[3] H. Wang *et al.*, "Groot: An Event-graph-based Approach for Root Cause Analysis in Industrial Settings," in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*, Melbourne, Australia, 2021, S. 419–429, doi: 10.1109/ASE51524.2021.9678708.

[4] S. Ghosh, M. Shetty, C. Bansal und S. Nath, "How to fight production incidents?," S. 126–141.

[5] Z. Chen *et al.*, "Towards intelligent incident management: why we need it and how we make it," S. 1487–1497.

[6] "Cost of Data Center Outages: January 2016," Ponemon Institute LLC, Data Center Performance Benchmark Series, Jan. 2016. Zugriff am: 13. Januar 2024.

[7] Y. Chen *et al.*, "Automatic Root Cause Analysis via Large Language Models for Cloud Incidents," Mai. 2023. [Online]. Verfügbar unter: http://arxiv.org/pdf/2305.15778.pdf

[8] ASReview. "Research | ASReview." Zugriff am: 15. Februar 2024. [Online.] Verfügbar: https://asreview.nl/research/

[9] T. Q. A. Lead. "New Relic Software Testing Tool In-Depth Review 2024." Zugriff am: 23. Januar 2024. [Online.] Verfügbar: https://theqalead.com/tools/new-relic-review/

[10] G. Yu, P. Chen, Y. Li, H. Chen, X. Li und Z. Zheng, "Nezha: Interpretable Fine-Grained Root Causes Analysis for Microservices on Multi-modal Observability Data," in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, San Francisco CA USA, S. Chandra, K. Blincoe und P. Tonella, Hg., 2023, S. 553–565, doi: 10.1145/3611643.3616249.

[11] R. Li, M. Du, Z. Wang, H. Chang, S. Mukherjee und E. Eide, "LongTale: Toward Automatic Performance Anomaly Explanation in Microservices," in *Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering*, 2022, S. 5–16, doi: 10.1145/3489525.3511675.

[12] D. Wang, Z. Chen, Y. Fu, Y. Liu und H. Chen, "Incremental Causal Graph Learning for Online Root Cause Analysis," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach CA USA, A. Singh et al., Hg., 2023, S. 2269–2278, doi: 10.1145/3580305.3599392.

[13] R. Ding *et al.*, "TraceDiag: Adaptive, Interpretable, and Efficient Root Cause Analysis on Large-Scale Microservice Systems," Okt. 2023. [Online]. Verfügbar unter: https://arxiv.org/pdf/2310.18740.pdf

[14] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang und X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Trans. Knowl. Data Eng.*, Jg. PP, Nr. 99, S. 1–20, 2024. doi: 10.1109/TKDE.2024.3352100. [Online]. Verfügbar unter: https://www.researchgate.net/publication/377318531_Unifying_Large_Language_Models_and_Knowledge_Graphs_A_Roadmap

[15] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao und Ningyu Zhang, *LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities*, 2023. [Online]. Verfügbar unter: https://www.researchgate.net/publication/370949054_LLMs_for_Knowledge_Graph_Construction_and_Reasoning_Recent_Capabilities_and_Future_Opportunities