applications behaviors [1, 2, 4, 19]. For example, authors. in [2] inject application-level events into kernel traces by executing a series of innocuous system calls for each high-level event of interest (e.g., the start and end of an RPC call). These system calls serve as synchronization points in the trace to merge high-level and low-level events. Also, in [4], Belkhiri et al. relied on vertical context propagation to inject high-level request identifiers into the kernel. The weakness of this approach is that it poses scalability challenges as it requires a system call each time the target application starts or completes the processing of a request. There are also numerous attempts to diagnose performance anomalies by applying statistics, graph theory, and trace comparison techniques on collected traces [8, 14, 22]. For instance, Huang et al. in [14] leverage the structure within the distributed traces to group similar traces and provide detailed statistics at each level of the trace hierarchy. Their tool can assist practitioners in identifying the relevant operations to focus on when debugging but cannot identify the cause of the issue automatically.

5 CONCLUSION

Microservices often complicate the debugging of unexpected latencies in application operations and pinpointing their root causes. This paper addresses this issue by proposing an innovative approach for diagnosing performance anomalies in microservice applications. Our approach leverages cross-layer tracing to enhance the granularity of observability, providing a multi-dimensional view that correlates system resource metrics with user requests. The use of sequential pattern mining enables the isolation of anomalous behavior patterns and facilitates the identification of their root causes. Our evaluations have not only confirmed the efficacy of our framework in identifying performance anomalies but also demonstrated its operational efficiency by maintaining minimal overhead.

As distributed systems evolve, diagnosing performance grows increasingly complex. Our contribution represents a step forward in mitigating this challenge by equipping developers and system operators with a tool capable of identifying and diagnosing performance issues without invasive instrumentation or prohibitive performance penalties. We expect that our findings will incite further research into optimizing distributed tracing infrastructures and developing even more sophisticated analysis techniques. Future work could explore the potential for real-time anomaly detection and automated remediation, which would enhance further the resilience and reliability of microservice-based applications.

REFERENCES

[1] Marcelo Amaral, Tatsuhiro Chiba, Scott Trent, Takeshi Yoshimura, and Sunyanan Choochotkaew. 2022. MicroLens: A Performance Analysis Framework for Microservices Using Hidden Metrics With BPF. *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)* 00 (2022), 230–240.

[2] Dan Ardelean, Amer Diwan, and Chandra Erdman. 2018. Performance Analysis of Cloud Applications. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, Vol. 00. USENIX Association, Renton, WA, 405–417.

[3] Emre Ates, Lily Sturmann, Mert Toslali, Orran Krieger, Richard Megginson, Ayse K. Coskun, and Raja R. Sambasivan. 2019. An automated, cross-layer instrumentation framework for diagnosing performance problems in distributed applications. *Proceedings of the ACM Symposium on Cloud Computing* (2019), 165–170.

[4] Adel Belkhiri, Ahmad Shahnejat Bushehri, Felipe Gohring de Magalhaes, and Gabriela Nicolescu. 2023. Transparent Trace Annotation for Performance Debugging in Microservice-oriented Systems (Work In Progress Paper). *International Conference on Performance Engineering (ACM/SPEC)* (2023), 25–32.

[5] Yang Cai, Biao Han, Jinshu Su, and Xiaoyan Wang. 2021. TraceModel: An Automatic Anomaly Detection and Root Cause Localization Framework for Microservice Systems. *2021 17th International Conference on Mobility, Sensing and Networking (MSN)* 00 (2021), 512–519.

[6] CNCF. 2023. OpenTelemetry: high-quality, ubiquitous, and portable telemetry to enable effective observability. https://opentelemetry.io/

[7] Mathieu Desnoyers and Michel R Dagenais. 2006. The lttng tracer: A low impact performance and behavior monitor for gnu/linux. In *OLS (Ottawa Linux Symposium)*, Vol. 2006. Citeseer, 209–224.

[8] Prem Devanbu, Myra Cohen, Tao Xie, and Liangfei Su. 2020. Graph-based trace analysis for microservice architecture understanding and problem diagnosis. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (2020).

[9] Ericsson. 2023. Trace Compass. http://tracecompass.org/

[10] Dan Feng, Steffen Becker, Nikolas Herbst, Philipp Leitner, Zheng Papadopoulos, Hyunseok Chang, Sarit Mukherjee, and Eric Eide. 2022. LongTale: Toward Automatic Performance Anomaly Explanation in Microservices. *International Conference on Performance Engineering (ACM/SPEC)* (2022), 5–16.

[11] Rodrigo Fonseca, George Porter, Randy H. Katz, and Scott Shenker. 2007. X-Trace: A Pervasive Network Tracing Framework. In *4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 07)*. USENIX Association, Cambridge, MA.

[12] Francis Giraldeau and Michel Dagenais. 2015. Wait Analysis of Distributed Systems Using Kernel Tracing. *IEEE Transactions on Parallel and Distributed Systems* 27, 8 (2015), 2450–2461.

[13] Jonatan Heyman, Joakim Hamrén, Carl Byström, and Hugo Heyman. 2023. Locust: An open-source load testing tool. https://locust.io/

[14] Lexiang Huang and Timothy Zhu. 2021. tprof: Performance profiling via structural aggregation and automated analysis of distributed systems traces. *Proceedings of the ACM Symposium on Cloud Computing* (2021), 76–91.

[15] Jaegertracing.io. 2023. Jaeger: Open Source, End-to-End Distributed Tracing. http://jaegertracing.io

[16] Madeline Janecek, Naser Ezzati-Jivan, and Seyed Vahid Azhari. 2021. Container Workload Characterization Through Host System Tracing. *2021 IEEE International Conference on Cloud Engineering (IC2E)* 00 (2021), 9–19.

[17] Jonathan Kaldor, Jonathan Mace, and Yee Jiun Song. 2017. Canopy: An End-to-End Performance Tracing And Analysis System. *Proceedings of the 26th Symposium on Operating Systems Principles* (2017), 34–50.

[18] Iman Kohyarnejadfard, Daniel Aloise, Seyed Vahid Azhari, and Michel R. Dagenais. 2022. Anomaly detection in microservice environments using distributed tracing data analysis and NLP. *Journal of Cloud Computing* 11, 1 (2022), 25.

[19] Cheryl Lee, Tianyi Yang, Zhuangbin Chen, Yuxin Su, and Michael R Lyu. 2023. Eadro: An End-to-End Troubleshooting Framework for Microservices on Multi-source Data. *arXiv* (2023). arXiv:2302.05092

[20] Chun Li and Jianyong Wang. 2008. Efficiently Mining Closed Subsequences with Gap Constraints. *Proceedings of the 2008 SIAM International Conference on Data Mining* (2008), 313–322.

[21] Ping Liu, Haowen Xu, Qianyu Ouyang, Rui Jiao, Zhekang Chen, Shenglin Zhang, Jiahai Yang, Linlin Mo, Jice Zeng, Wenman Xue, and Dan Pei. 2020. Unsupervised Detection of Microservice Trace Anomalies through Service-Level Deep Bayesian Networks. *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)* 00 (2020), 48–58.

[22] Lun Meng, Feng Ji, Yao Sun, and Tao Wang. 2021. Detecting anomalies in microservices with execution trace comparison. *Future Generation Computer Systems* 116 (2021), 291–301.

[23] Inc. NGINX. 2023. Bank of Sirius. https://github.com/nginxinc/bank-of-sirius

[24] Tim Sherwood, Emery Berger, Christos Kozyrakis, Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. 2021. Sage: practical and scalable ML-driven performance debugging in microservices. *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (2021), 135–151.

[25] Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, and Chandan Shanbhag. 2010. *Dapper, a Large-Scale Distributed Systems Tracing Infrastructure.* Technical Report. Google, Inc.

[26] Mert Toslali, Emre Ates, Alex Ellis, Zhaoqi Zhang, Darby Huye, Lan Liu, Samantha Puterman, Ayse K. Coskun, and Raja R. Sambasivan. 2021. Automating instrumentation choices for performance problems in distributed applications with VAIF. *Proceedings of the ACM Symposium on Cloud Computing* (2021), 61–75.

[27] Dean M. Tullsen and Brad Calder. 1998. *Computing along the critical path.* Technical Report. Technical report, University of California, San Diego.

[28] Zhizhou Zhang, Murali Krishna Ramanathan, Prithvi Raj, Abhishek Parwal, Timothy Sherwood, and Milind Chabbi. 2022. CRISP: Critical Path Analysis of Large-Scale Microservice Architectures. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. USENIX Association, Carlsbad, CA, 655—672.

[29] Zipkin.io. 2022. Zipkin. https://zipkin.io