Closing the Loop: Building Self-Adaptive Software for Continuous Performance Engineering

Marin Litoiu

Electrical Engineering and Computer Science Department, School of IT

York University

Totonto, Ontario, Canada

mlitoiu@yorku.ca

ABSTRACT

Cloud computing and cloud-native platforms have rendered runtime environments more malleable. Simultaneously, the growing demand for flexible and agile software applications and services has driven the emergence of self-adaptive architectures. These architectures, in turn, facilitate software performance modeling, tuning, optimization, and scaling in a continuous manner, blurring the boundary between development-time and run-time. Self-adaptive software employs feedback loop controllers inspired by control theory or variations of the Monitoring-Analysis-Planning-Acting (MAPE) architecture. Whether implemented in a centralized or decentralized manner, most controllers utilize performance models that are learned or tuned at run-time. This shift implies that software is designed to be observable and controllable during execution, presupposing the co-design of software applications and their runtime controllers.

This talk commences with a succinct overview of the evolution of self-adaptive software, accentuating key milestones along the journey. Subsequently, recent advancements in software performance modeling at runtime and the role of learning-enabled performance management during software operation are presented.

Two recent works are highlighted: one focusing on constructing robust performance models to sustain continuous operation and deployment of cloud-native software, and the other on utilizing multimodal models for performance anomaly detection. The former supports cloud operations like continuous deployment of co-located applications, migration, consolidation of services, or scaling in response to workloads or interferences. The latter is tailored to support performance anomaly detection, localization, and identification of root causes, facilitating swift remediation of faults using generative AI. The final segment of the talk delves into current challenges in developing self-adaptive systems, presenting insights from a recent survey on the state of self-adaptive software in the industry and the challenges perceived by practitioners.

CCS CONCEPTS

•Software and its engineering •Software and its engineering~Software organization and properties~Extra-functional properties~Software performance

KEYWORDS

Self-adaptive software systems, self-optimization, software performance, performance models, cloud computing, machine learning, generative AI

BIOGRAPHY

Marin Litoiu is a Professor of Software Engineering in the Department of Electrical Engineering and Computer Science and in the School of Information Technology, York University. He is also a Fellow of the Canadian Academy of Engineering. Dr. Litoiu leads the Adaptive Software Research Lab and focuses on making large software systems more versatile, resilient, energy-efficient, self-healing and self-optimizing. His research won many awards including the IBM Canada CAS Research Project of the Year Award, the IBM CAS Faculty Fellow of the Year Award for his "impact on IBM people, processes and technology," three Best Paper Awards and two Most Influential Paper Awards. Prior to joining York University, Dr. Litoiu was a Research Staff member with the Centre for Advanced Studies in the IBM Toronto Lab where he led the research programs in software engineering and autonomic computing. He received the Canada NSERC Synergy Award for Innovation in recognition for these collaborative university/industry activities. He was also

recipient of the IBM Outstanding Technical Contribution Award for his research vision on Cloud Computing. Dr. Litoiu is one of the founders of the SEAMS Symposium series—ACM/IEEE Software Engineering for Adaptive and Self-Managing Systems. Dr. Litoiu is also the Scientific Director of "Dependable Internet of Things Applications (DITA)," an NSERC CREATE program.

ACKNOWLEDGMENTS

REFERENCES

[1] Sarda Komal, Namrud Zakeya, Rouf Raphael, Ahuja Harit, Rasolroveicy Mohammadreza, Litoiu Marin, Shwartz Larisa, and Watts Ian. 2023. ADARMA Auto-Detection and Auto-Remediation of Microservice Anomalies by Leveraging Large Language Models. In Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering (CASCON '23). IBM Corp., USA, 200–205.

[2] T. Zheng, C. M. Woodside and M. Litoiu, "Performance Model Estimation and Tracking Using Optimal Filters," in *IEEE Transactions on Software Engineering*, vol. 34, no. 3, pp. 391-406, May-June 2008, doi: 10.1109/TSE.2008.30.

[3] Y. Rouf, J. Mukherjee and M. Litoiu, "Towards a Robust On-line Performance Model Identification for Change Impact Prediction," *2023 IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Melbourne, Australia, 2023, pp. 68-78, doi: 10.1109/SEAMS59076.2023.00018.

[4] D. Weyns, M.Litoiu *et al.*, "Towards Better Adaptive Systems by Combining MAPE, Control Theory, and Machine Learning," *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Madrid, Spain, 2021, pp. 217-223, doi: 10.1109/SEAMS51251.2021.00036.

[5] D. Weyns, M. Litoiu et al.. 2023. Self-Adaptation in Industry: A Survey. ACM Trans. Auton. Adapt. Syst. 18, 2, Article 5 (June 2023), 44 pages. https://doi.org/10.1145/3589227

[6] R. Rouf, M. Rasolveicy, M Litoiu et al., InstantOps: A Joint Approach to System Failure Prediction and Root Cause Identification in Microservices Cloud-Native
Applications, Proceedings of ACM/SPEC ICPE 2024, London, UK, May 2024.

[7] K. Sarda, Z. Narud, M. Litoiu et al. KubePlaybook: A Repository of Ansible Playbooks for Kubernetes Auto-Remediation with LLMs, Proceedings of ACM/SPEC ICPE 2024, London, UK, May 2024.

[8] de Lemos, R. *et al.* (2013). Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds) Software Engineering for Self-Adaptive Systems II. Lecture Notes in Computer Science, vol 7475. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35813-5_1