Received: 14 November 2023 | Accepted: 18 November 2023

DOI: 10.1002/spe.3298

EDITORIAL

Special issue on efficient management of microservice-based systems and applications

The advent of microservice architecture marks a transition from conventional monolithic applications to a landscape of loosely linked, lightweight, and autonomous microservice components. The primary objective is to ensure strong environmental uniformity, portability across various operating systems, and robust resource isolation. Leading cloud service providers such as Amazon, Microsoft, Google, and Alibaba have widely embraced microservices within their infrastructures. This adoption is geared toward automating application management and optimizing system performance. Consequently, addressing the automation of tasks like deployment, maintenance, auto-scaling, and networking of microservices becomes pivotal. This underscores the importance of efficient management of systems and applications built on microservices as a critical research challenge.

Efficient management methods must not only ensure the quality of service (QoS) across multiple microservices units (containers) but also provide greater control over individual components. However, the dynamic and varied nature of microservice applications and environments significantly amplifies the complexity of these management approaches. Each microservice unit can be deployed and operated independently, catering to distinct functionalities and business objectives. Furthermore, microservices can interact and combine through lightweight communication techniques to form a complete application. The expanding scale of microservice-based systems and their intricate interdependencies pose challenges in terms of load distribution and resource management at the infrastructure level. Furthermore, as cloud workloads surge in resource demands, bandwidth consumption, and QoS requirements, the traditional cloud computing environment extends to fog and edge infrastructures that are in close proximity to end users. As a result, current microservice management approaches need further enhancement to address the mounting resource diversity, application distribution, workload profiles, security prerequisites, and scalability demands across hybrid cloud infrastructures.

Keeping this in mind, this special issue addressed some of the aspects related to efficient management of microservice-based systems and applications with the focus on various challenges faced, and promising solutions to address such challenges by using software engineering, machine learning and deep learning techniques. We have received 21 submissions in this issue, and we accepted six high-quality submissions for publication after a rigorous review process with at least three reviewers for each paper. The authors are from diverse countries, including the USA, China, UK, Germany, India, Brazil, etc. Each of the accepted papers is summarized as follows.

In the first article, Batista et al.[1] presented two strategies for handling asynchronous workloads associated with tax integration in a multi-tenant microservice architecture specific to the company's context. The initial approach utilizes polling, employing a queue as a distributed lock. The second approach, named the single active consumer, adopts a push-based technique, leveraging the message broker's logic for message delivery. These methodologies are designed to optimize resource allocation in scenarios involving an increasing number of container replicas and tenants.

In the second article, Kumar et al.[2] introduced a resource allocation model designed to enhance the QoS in microservice deployment. Utilizing a Fine-tuned Sunflower Whale Optimization Algorithm, the model strategically deploys container-based services on physical machines, optimizing their execution capacity by efficiently utilizing CPU and memory resources. The primary objective of this proposed technique is to achieve an efficient distribution of workload, preventing resource wastage and ultimately enhancing QoS parameters.

In the third paper, Zhu et al.[3] introduced RADF, a semi-automatic approach for decomposing a monolith into serverless functions by analyzing the inherent business logic present in the application's interface. The proposed method employs a two-stage refactoring strategy, initially performing a coarse-grained decomposition followed by a fine-grained one. This approach streamlines the decomposition process into smaller, more manageable steps, providing adaptability to generate a solution at either the microservice or function level.

In the fourth paper, Würz et al.[4] identified the principal tasks and subtasks of the application for the purpose of partitioning. Subsequently, they outlined the program flow to ascertain which application tasks could be transformed into functions and elucidated their interdependencies. In the concluding step, they precisely specified individual functions and, if necessary, consolidated those deemed excessively small to mitigate communication overhead or maintenance challenges. In contrast to the previous methodologies, their approach is universally applicable to applications of varying sizes, ensuring that the resulting functions are appropriately sized for efficient execution within a Function as a Service environment.

In the fifth paper, Zhong et al.[5] introduced DOMICO, a methodology designed to assess the conformance between a domain model and its implementation. This conformance is established through the formalization of eight common structural patterns in domain modeling and their representations in both the models and the associated source codes. Utilizing this formalization, the approach can pinpoint discrepancies, such as deviations, omissions, and modifications, concerning pattern elements. Furthermore, DOMICO is capable of detecting potential violations of 24 compliance rules imposed by these patterns.

In the sixth paper, Zhang et al.[6] proposed a chunk reuse mechanism aimed at efficiently identifying node-local duplicate data during container updates. This mechanism contributes to a reduction in the volume of data transmission needed for image building. The authors manage the chunk reuse mechanism process for both cloud and remote-cloud nodes, ensuring that the associated resource overhead from the preparation of container update data and image reconstruction remains within acceptable thresholds.

We hope that the accepted contributions in this special issue will help readers of the journal and the wider research community gain knowledge on the presented research challenges, techniques, and solutions. We also hope that it will encourage them to further work on different aspects of efficient management of microservice-based clusters.

ACKNOWLEDGMENTS

We would like to express our gratitude to the editorial board members for granting us the opportunity to conduct a special issue in Software: Practice and Experience. We would also like to extend our appreciation to the administrative staff, reviewers, and most importantly, the authors, for their invaluable assistance and contributions in the successful organization of this issue.

Minxian Xu[1]
Schahram Dustdar[2]
Massimo Villari[3]
Rajkumar Buyya[4]

[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[2]Distributed Systems Group, Vienna University of Technology, Vienna, Austria
[3]Department of MIFT, University of Messina, Messina, Italy
[4]CLOUDS Lab, School of Computing and Information Systems, University of Melbourne, Melbourne, Australia

Correspondence
Minxian Xu, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.
Email: mx.xu@siat.ac.cn

ORCID

Schahram Dustdar https://orcid.org/0000-0001-6872-8821
Massimo Villari https://orcid.org/0000-0001-9457-0677
Rajkumar Buyya https://orcid.org/0000-0001-9754-6496

REFERENCES

1. Batista C, Morais F, Cavalcante E, Batista T, Proença B, Rodrigues Cavalcante WB. Managing asynchronous workloads in a multi-tenant microservice enterprise environment. *Softw: Pract Exp*. 2024;54(2):334-359. doi:10.1002/spe.3278
2. Kumar M, Samriya JK, Dubey K, Gill SS. QoS-aware resource scheduling using whale optimization algorithm for microservice applications. *Softw: Pract Exp*. 2024;54(4):546-565. doi:10.1002/spe.3211
3. Zhu L, Tamburri DA, Casale G. RADF: Architecture decomposition for function as a service. *Softw: Pract Exp*. 2024;54(4):566-594. doi:10.1002/spe.3276

4. Würz HM, Krämer M, Kaster M, Kuijper A. Migrating monolithic applications to function as a service. *Softw: Pract Exp*. 2024;54(2):149-167. doi:10.1002/spe.3263
5. Zhong C, Zhang H, Huang H, Chen Z, Li C, Li S. DOMICO: checking conformance between domain models and implementations. *Softw: Pract Exp*. 2024;54(4):595-616. doi:10.1002/spe.3272
6. Zhang H, Lin W, Xie R, Li S, Dai Z, Wang JZ. An optimal container update method for edge-cloud collaboration. *Softw: Pract Exp*. 2024;54(4):617-634. doi:10.1002/spe.3232