The Cloudification Perspectives of Search-based Software Testing

Diego Martin
Zurich University of Applied Science (ZHAW)
Winterthur, Switzerland
marg@zhaw.ch

Sebastiano Panichella
Zurich University of Applied Science (ZHAW)
Winterthur, Switzerland
panc@zhaw.ch

***Abstract*—To promote and sustain the future of our society, the most critical challenge of contemporary software engineering and cloud computing experts are related to the efficient integration of emerging cloudification and DevOps practices in the development and testing processes of modern systems. In this context, we argue that SBST can play a critical role in improving testing practices and automating the verification and validation (V&V) of cloudification properties of Cloud Native Applications (CNA). Hence, in this paper, we focus on the untouched side of SBST in the cloud field, by discussing (1) the testing challenges in the cloud research field and (2) summarizing the recent contributions of SBST in supporting development practices of CNA. Finally, we discuss the emerging research topics characterizing the cloudification perspectives of SBST in the cloud field.**

***Keywords*-Cloud Native Applications, Search-based Software Testing, Test Suite Generation**

I. Introduction

The first work on Search-Based Software Testing (SBST) appeared in 1976 [1] and in the last decade the SBST research field reached a high maturity, with several research work and tools [2]–[5] aimed at supporting test data generation and test suite quality assessment [6]–[9].

The most critical challenge of software engineering and cloud computing experts is related to the efficient integration of emerging cloudification [10] and DevOps practices in the development and testing processes of modern systems. This put in place the need of solutions ensuring, quantifying, and verifying the elastic scalability (i.e., adjusting their capacity by adding or removing resources) and resiliency (i.e., anticipating failures and fluctuation) [11] of the software/hardware (micro)services composing the systems.

In the cloud field, SBST strategies have been recently experimented for task scheduling [12] and service composition [13]. We argue that SBST can play a critical role in the supporting the verification and validation (V&V) of cloud native applications. Hence, in this paper, we summarize the main challenges and opportunities on the untouched side of the SBST in the cloud field.

II. Analysis and Discussion

A. Cloud Testing Literature Insights

Cloud-native applications (CNAs) are *"distributed, elastic and horizontal-scalable systems composed of (micro)services which isolates states in a minimum of stateful components"* [14]. Each self-contained deployment unit of CNAs is designed according to cloud-focused design patterns and operated on a self-service elastic platform. This means that CNAs are supposed to be ***composable***, ***decoupled***, ***elastic*** and ***resilient*** [14].

Even thought most of these *properties* have been investigated/analyzed in the literature [15], [16], there are still open problems/challenges where SBST could contribute:

1) V&V of CNA properties: Provide automated solutions for V&V of CNA properties like *composition, decoupling, elasticity and resilience.*

2) CNAs Adaptations: Provide tools for V&V of cloud migrations or migration between cloud providers.

3) Microservices Evolution: Support developers with tools providing an architectural view of microservices evolution, with V&V on the side-effects of specific changes in microservices composition and orchestration.

4) Local v.s. Global V&V: Provide automated V&V of microservices evolutions that verify the behaviour of each microservice in isolation (local level) as well as the global microservices *behaviour* and *interactions* in the system.

5) Execution Time: Execution of test suites should be efficient and effective, and should encapsulate advanced coverage criteria for microservices based architecture.

B. Literature review of SBST for the cloud

SBST literature in the cloud field focused on the optimization of testing frameworks [17], [18] and the stress of basic CNA properties (elasticity [19] and resilience [20]), with some work analyzing the challenges of cloud migrations [21], and the possibilities of using combinatorial testing [22]. However, some of the the challenges previously identified are completely untouched.

We conducted a literature review on SBST papers published in the last 5 years (period 2014-2019), focusing on the works that contributed to the cloud research field.

The literature review has been performed by using DBLP as main source of information. Specifically, in the period analyzed, the selection of papers in DBLP was performed by using specific sets of search keywords, according to the following logic formula: $\text{CLOUD} \wedge (\text{GENETIC} \vee \text{COMBINATORIAL} \vee \text{SIMULATED ANNEALING} \vee \text{TABU SEARCH} \vee \text{PROFILING} \vee \text{SLICING} \vee \text{COVERAGE} \vee \text{COEVOLUTION} \vee \text{MUTATION} \vee \text{MORPH} \vee \text{HEALING} \vee \text{SELF REPAIR} \vee \text{HILL CLIMBING} \vee \text{SEARCH BASED})$.

The aforementioned filter resulted in only **6** papers relevant to SBST applications in the cloud field from the **396**

initial papers. Replication package available online[1].

C. SBST cloud future perspectives

In this section, we discuss how SBST is a potential solution to the challenges identified in the cloud testing field.

1) V&V of CNA properties & Local v.s. Global V&V: Test case generation for the V&V of CNA properties is still a widely unexplored field, and metrics and tools should be designed in order to guide the generation and the quality assessment of corresponding tests. Finally, to ensure a local and global V&V of CNA applications, properties, and microservices composition/interactions, we envision SBST strategies, encapsulating smart mechanisms able to assess the inputs that most impact, affect, interacts with the coordination and evolution of the microservices architecture.

2) CNAs Adaptations: SBST techniques and their use for V&V of CNA properties has a perfect matching for the analytic support to *CNA Adaptation/Migration* challenges.

3) Microservices Evolution & System Coverage: The use of profiling and slicing dynamic analysis techniques can be combined with SBST strategies to understand/monitor how CNA code fragments behave, and to identify potential architectural flaws or flow bottlenecks in the execution time. This can give the developers a profound understanding/view of the evolution of a system as a whole.

4) Execution Time: SBST strategies, from Simulated Annealing, Hill Climbing to Ant Colonies techniques, are characterized by the capabilities to lead to autonomous optimizations, with high potential in helping developers achieve the typical execution-time requirements of CNA.

D. Position

The authors posit that SBST techniques can play a major role in the upcoming future of cloud testing, with automated strategies based on CNA properties and practices, enabling dynamic failures identification and fixing.

References

[1] W. Miller and D. L. Spooner, "Automatic generation of floating-point test data," *IEEE Trans. Software Eng.*, vol. 2, no. 3, pp. 223–226, 1976.

[2] G. Fraser and A. Arcuri, "Evosuite: automatic test suite generation for object-oriented software," in *ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 2011, pp. 416–419.

[3] A. Panichella, F. M. Kifetew, and P. Tonella, "Reformulating branch coverage as a many-objective optimization problem," in *International Conference on Software Testing, Verification and Validation*, 2015, pp. 1–10.

[4] ——, "Automated test case generation as a many-objective optimisation problem with dynamic selection of the targets," *IEEE Trans. Software Eng.*, vol. 44, no. 2, pp. 122–158, 2018.

[5] M. Khari and P. Kumar, "An extensive evaluation of search-based software testing: a review," *Soft Computing*, vol. 23, no. 6, pp. 1933–1946, Mar 2019.

[6] P. McMinn, "Search-based software testing: Past, present and future," in *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 2011, pp. 153–163.

[7] M. Harman, S. A. Mansouri, and Y. Zhang, "Search-based software engineering: Trends, techniques and applications," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 11:1–11:61, 2012.

[8] S. Panichella, A. Panichella, M. Beller, A. Zaidman, and H. C. Gall, "The impact of test case summaries on bug fixing performance: an empirical investigation," in *Proceedings of the 38th International Conference on Software Engineering, ICSE*, 2016, pp. 547–558.

[9] S. Panichella, "Summarization techniques for code, change, testing, and user feedback (invited paper)," in *Workshop on Validation, Analysis and Evolution of Software Tests, VST@SANER 2018*, 2018, pp. 1–5.

[10] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Computing*, vol. 1, no. 2, pp. 142–157, 2013.

[11] M. R. Namjoo, A. Keramati, S. A. Torabi, and F. Jolai, "Quantifying the resilience of cloud-based manufacturing composite services," *IJCAC*, vol. 8, no. 4, pp. 88–117, 2018.

[12] N. Gobalakrishnan and C. Arun, "A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing," *Comput. J.*, vol. 61, no. 10, pp. 1523–1536, 2018.

[13] F. Seghir and A. Khababa, "A hybrid approach using genetic and fruit fly optimization algorithms for qos-aware cloud service composition," *J. Intelligent Manufacturing*, vol. 29, no. 8, pp. 1773–1792, 2018.

[14] N. Kratzke and P. Quint, "Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study," *Journal of Systems and Software*, vol. 126, pp. 1–16, 2017.

[15] A. R. Cavalli, T. Higashino, and M. Núñez, "A survey on formal active and passive testing with applications to the cloud," *Annales des Télécommunications*, vol. 70, no. 3-4, pp. 85–93, 2015.

[16] P. Chawla, I. Chana, and A. Rana, "Empirical evaluation of cloud-based testing techniques: a systematic review," *ACM SIGSOFT Software Engineering Notes*, vol. 37, no. 3, pp. 1–9, 2012.

[17] P. C. Cañizares, A. Núñez, and M. G. Merayo, "Mutomvo: Mutation testing framework for simulated cloud and HPC environments," *Journal of Systems and Software*, vol. 143, pp. 187–207, 2018.

[18] A. Al-Ghuwairi, H. Eid, M. Aloran, Z. Salah, A. H. Baarah, and A. A. Al-oqaily, "A mutation-based model to rank testing as a service (taas) providers in cloud computing," in *International Conference on Internet of Things and Cloud Computing*, 2016, pp. 18:1–18:5.

[19] A. Alourani, M. A. N. Bikas, and M. Grechanik, "Search-based stress testing the elastic resource provisioning for cloud-based applications," in *Search-Based Software Engineering*, 2018, pp. 149–165.

[20] W. F. F. Cardoso and E. Martins, "Using a search and model based framework to improve robustness tests in cloud platforms," in *Symposium on Systematic and Automated Software Testing*, 2018, pp. 67–76.

[21] Y. Magid, R. Tzoref-Brill, and M. Zalmanovici, "Coverage-based metrics for cloud adaptation," in *International Workshop on Quality-Aware DevOps*, 2016, pp. 1–6.

[22] W. Tsai and G. Qi, *Combinatorial Testing in Cloud Computing*, ser. Springer Briefs in Computer Science.

[1] https://github.com/GoDieNow/CloudSBSTPerspectives