Test Suite Reduction in Idempotence Testing of Infrastructure as Code

Abstract

Infrastructure as Code, which uses machine-processable code for managing, provisioning, and configuring computing infrastructure, has been attracting wide attention. In its application, the idempotence of the code is essential: the system should converge to the desired state even if the code is repeatedly executed possibly with failures or interruptions. Previous studies have used testing or static verification techniques to check whether the code is idempotent or not. The testing approach is impractically time-consuming, whereas the static verification approach is not applicable in many practical cases in which external scripts are used. In this paper, we present a method for efficiently checking idempotence by combining the testing and static verification approaches. The method dramatically decreases the number of test cases used to check code including external scripts by applying the static verification approach.

Keywords

Execution Time, Outgoing Edge, Source Language, Satisfiability Modulo Theory, Incoming Edge

Acknowledgment

References

1. Burgess, M.: Testable system administration. Commun. ACM **54**(3), 44–49 (2011).

2. Caroprese, L., Trubitsyna, I., Truszczynski, M., Zumpano, E.: The view-update problem for indefinite databases (2012). CoRR, abs/1205.4655.

3. Cartaxo, E.G., Machado, P.D.L., Oliveira Neto, F.G.: On the use of a similarity function for test case selection in the context of model-based testing. Softw. Test. Verif. Reliab. **21**(2), 75–100 (2011).

4. Chef Software, Inc.: Chef Embrace DevOps - Chef. https://www.chef.io/

5. Chef Software, Inc.: Welcome - The Resource for Chef cookbooks - Chef Supermarket. https://supermarket.chef.io/

6. Chen, T.Y., Lau, M.F.: A simulation study on some heuristics for test suite reduction. Inf. Softw. Technol. **40**(13), 777–787 (1998).

7. Collard, J.M., Gupta, N., Shambaugh, R., Weiss, A., Guha, A.: On static verification of puppet system configurations (2015). CoRR, abs/1509.05100.

8. Couch, A., Sun, Y.: On the algebraic structure of convergence. In: Brunner, M., Keller, A. (eds.) DSOM 2003. LNCS, vol. 2867, pp. 28–40. Springer, Heidelberg (2003). doi: 10.1007/978-3-540-39671-0_4

9. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). doi: 10.1007/978-3-540-78800-3_24

10. Foster, J.N., Greenwald, M.B., Moore, J.T., Pierce, B.C., Schmitt, A.: Combinators for bidirectional tree transformations: a linguistic approach to the view-update problem. ACM Trans. Program. Lang. Syst. (TOPLAS) **29**(3), 17 (2007).

11. Heavy Water Operations, LLC (OR): Welcome to Test Kitchen - KitchenCI. http://kitchen.ci/

12. Hummer, W., Rosenberg, F., Oliveira, F., Eilam, T.: Testing idempotence for infrastructure as code. In: Eyers, D., Schwan, K. (eds.) Middleware 2013. LNCS, vol.

8275, pp. 368–388. Springer, Heidelberg (2013). doi: [10.1007/978-3-642-45065-5_19](10.1007/978-3-642-45065-5_19)

13. Kolokoltsov, V., Maslov, V.P.: Idempotent Analysis and Its Applications. Mathematics and Its Applications. Springer, Netherlands (1997).

14. Lin, J.-W., Huang, C.-Y.: Analysis of test suite reduction with enhanced tie-breaking techniques. Inf. Softw. Technol. **51**(4), 679–690 (2009).

15. Nelson-Smith, S.: Test-Driven Infrastructure with Chef: Bring Behavior-Driven Development to Infrastructure as Code. O'Reilly Media Inc., Sebastopol (2013).

16. 16. Puppet: Puppet - the shortest path to better software. https://puppet.com/

17. Red Hat, Inc.: Ansible is Simple IT Automation. https://www.ansible.com/

18. Vaysburg, B., Tahat, L.H., Korel, B.: Dependence analysis in reduction of requirement based test suites. In: Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2002, pp. 107–111. ACM, New York (2002).

19. Wittig, A., Wittig, M.: Amazon Web Services in Action, 1st edn. Manning Publications Co., Greenwich (2015).

20. Zhong, H., Zhang, L., Mei, H.: An experimental study of four typical test suite reduction techniques. Inf. Softw. Technol. **50**(6), 534–546 (2008).