Testing in Big Data: An Architecture Pattern for a Development Environment for Innovative, Integrated and Robust Applications

Daniel Staegemann[1], Johannes Hintsch[1] and Klaus Turowski[1]

[1] Otto-von-Guericke University, Faculty of Computer Science, Magdeburg, Germany
{daniel.staegemann,johannes.hintsch,klaus.turowski}@ovgu.de

Abstract. Big Data is a crucial pillar for many of today's newly emerging business models. Areas of application range from consumer analysis over medicine to fraud detection. All of those domains require reliable software. Even though imperfect results are accepted in Big Data software, bugs and other defects can have drastic consequences. Therefore, in this paper, the software engineering sub discipline of testing is addressed. Big Data exhibits characteristics which differentiate its processing software from those that process traditional workloads. Consequently, an architecture pattern for testing that can be integrated into development environments for Big Data software is proposed. The paper features a detailed description of the artifact as well as a preliminary plan for evaluation.

Keywords: Big Data, Testing, Design Science, Software Engineering.

1 Introduction

Big data ranked the top-most important area of IT investments throughout the past five consecutive years [1]. Firms use data to get new insights (e.g., about customers' purchasing preferences) or to make decisions (e.g., in credit card fraud management). Even though the potential is high [2], companies are struggling to cope with the implicated challenges [3–5]. As an important part of the software development process "Software testing is a process, or a series of processes, designed to make sure computer code does what it was designed to do and, conversely, that it does not do anything unintended" [6]. Therefore, all activities that are supposed to determine the congruence of a program and its pre-defined requirements can be deemed software testing. The necessity to rigorously test software stems from the potential harm, that even seemingly little mistakes in the software can cause [7]. Architectures are the "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" [8]. Big Data, as a new paradigm, challenges the architecture of traditional software engineering environments, particularly in testing [9]. This is due to the properties of Big Data, often characterized as the four "V"s. Those are volume (amount of data), variety (different sources of data), velocity (rate of the dataflow) and variability

14th International Conference on Wirtschaftsinformatik,
February 24-27, 2019, Siegen, Germany

(change of data characteristics). These characteristics overstrain traditional data architectures and require new techniques, like the usage of horizontal scaling, to efficiently handle the respective datasets. Those challenges are also reflected in the related testing necessities [10].
To accommodate those necessities we follow the Design Science Research [11] paradigm to outline a testing architecture to support the development of big data applications. The focus is on domain specific applications that facilitate investigating the meaning of data and the relationships between different data. So far, social media analysis is deemed a promising domain for investigation.

2 Related Work

The diversity of different preliminary works in the existing literature reflects the complexity as well as the relevancy of the topic. It ranges from general descriptions of problem areas that also require testing [12], concepts on how to benchmark or test in the area of Big Data [13–16] and on the challenges of quality assurance [17] to more concrete approaches like an implementation for dataless testing [18]. There is however not a universally optimal solution yet, resulting in a need for further research.

3 Artifact

As mentioned beforehand, there are significant differences between software solutions for traditional data and those for Big Data. This results in additional challenges that need to be considered in the testing process as well as in the corresponding architecture of a software engineering environment. The three most notable challenges for testing in Big Data are the following. These were derived from literature [19–22] and from discussions with two experts.

- Difference 1: In contrast to traditional software in Big Data applications non-functional properties (like the ability to handle high volume and velocity) have a higher importance [10].
- Challenge 1: Huge amounts of varying data are required to test non-functional properties.
- Difference 2: Data are often heterogeneous (variety, variability) and the data quality is often poor [16].
- Challenge 2: Necessity to test the clearing and converting of source data.
- Difference 3: Due to the use-cases there is often a higher difficulty to determine if the tested system delivers optimal results [17].
- Challenge 3: The system is drafted to tackle situations that are complex in terms of data and could therefore not be handled with traditional technology, for this reason there is often no known set of inputs and matching outputs. (oracle problem) [23].

To provide maximum value, an architecture for testing Big Data systems should offer solutions for all of the mentioned challenges. Since the reviewed existing approaches were considered not sufficient in light of those challenges, the proposed one was created from scratch.

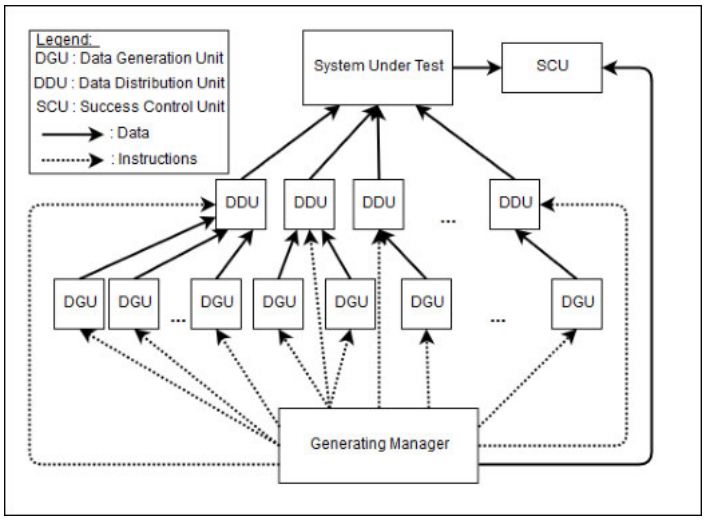

Figure 1. Architecture pattern for testing in Big Data development environments

The proposed architecture pattern, shown in Figure 1, consists of several elements, that aim to fulfill the identified requirements, when combined, to extensively stress the System under Test (SUT). A Generating Manager (GM) controls the whole process and steers the Data Generating Units (DGU) as well as the Data Distribution Units (DDU). If needed the GM can also create and terminate DGUs and DDUs. It is currently investigated, if an algorithm based on MapReduce [24] might be suitable for organizing the test procedure. The Success Control Unit (SCU) monitors the test, comparing the information sent by the GM with the results of the SUT. This allows for a real time monitoring of the performance of the SUT, regarding functional as well as non-functional aspects. The DGUs are each specialized on outputting one type of data (e.g. Twitter posts, reviews) and if needed specific characteristics (e.g. incompleteness, conflicting statements). This allows to choose the best possible solution for each creation sub-task instead of being bound to a solution that delivers acceptable but possibly suboptimal test data for all cases. This approach aims at testing the SUT's clearing, converting and processing of source data by feeding it data of varying type and quality, therefore tackling challenge 2.

Each DGU can generate data from scratch, by recreating existing data patterns, or outputs data that are provided by existing databases or data scientists. For this purpose it is given instructions by the GM. It is possible to have several DGUs with the same characteristics to achieve a higher rate of data generation. It is also feasible to create DGUs that are only providing data corresponding to the pattern the SUT is supposed to detect, while other DGUs are creating "decoy data" that does not comply. The

chosen approach targets an easy assessment regarding the detection rate of the SUT. This is because the GM knows which DGU's data are supposed to be detected by the SUT, therefore enabling the SCU as a test-oracle, addressing challenge 3. The DDUs are each devoted to one type of data, therefore utilizing the possible benefits of specialization. They are forwarded the data directly by the DGUs assigned to them by the GM. In the DDUs a buffer of data can be created for further use. When ordered by the GM, the DDUs send their data to the SUT, using the requested pattern, volume and velocity, utilizing the buffered data if needed, taking on challenge 1.

4 Evaluation

The evaluation follows the pattern proposed by Sonnenberg and vom Brocke [25]. EVAL 1 explores if the research and the accompanying creation of an artifact are justified or unnecessary. This step is included in the publication at hand. The general need for research in the outlined topic was illustrated, experts and relevant literature were included in the derivation of significant challenges and those were subsequently the foundation of the taken design decisions. This results in the hypothesis that the proposed architecture constitutes an improvement compared to existing approaches. EVAL 1 can therefore be deemed as completed. EVAL 2 focuses the feasibility and practicability of the suggested approach. It will use logical reasoning, comparing the challenges and the solutions, provided by the artifact, as well as an analysis to verify if the chosen test organization algorithm terminates and expert interviews, e.g. concerning expectable performance, to judge the feasibility of the developed architecture and to remedy possible flaws in the architecture or the algorithm. The prototype of the artifact itself and its testing constitute EVAL 3. Once the concept is implemented in real-life scenarios, a case study and further expert interviews are planned (EVAL 4). An overview of these described steps is depicted in Table 1.

Table 1. Evaluation Plan

Evaluation Steps	*Description*	*Status*
EVAL 1	This publication	Completed
EVAL 2	Logical reasoning and expert interview	Planned
EVAL 3	SAP HANA and OpenStack based prototyping	Planned
EVAL 4	Case study and expert interview	Planned

5 Conclusion

Big Data poses new challenges compared to traditional software engineering. The same applies to the corresponding testing. As a consequence there is currently no universally applied approach for testing Big Data systems. Using the modular artifact introduced in this publication provides possible solutions for those challenges of testing Big Data applications, while still respecting the potential uniqueness of individual projects and the belonging test scenarios.

References

1. Kappelman, L., Johnson, V., Maurer, C., McLean, E., Torres, R., David, A., Nguyen, Q.: The 2017 SIM IT Issues and Trends Study. MIS Quarterly Executive 17, 53–88 (2018)
2. McAfee, A., Brynjolfsson, E.: Big Data: The Management Revolution: Exploiting vast new flows of information can radically improve your company's performance. But first you'll have to change your decision-making culture. Harvard Business Review 91, 1–9 (2012)
3. Lee, I.: Big data: Dimensions, evolution, impacts, and challenges. Business Horizons 60, 293–303 (2017)
4. Yang, C., Huang, Q., Li, Z., Liu, K., Hu, F.: Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth 10, 13–53 (2017)
5. Tiwari, S., Wee, H.M., Daryanto, Y.: Big data analytics in supply chain management between 2010 and 2016: Insights to industries. Computers & Industrial Engineering 115, 319–330 (2018)
6. Myers, G.J., Badgett, T., Sandler, C.: The art of software testing. J. Wiley & Sons, Hoboken, N.J (2011)
7. Patton, R.: Software testing. SAMS, Indianapolis (2001)
8. ISO/IEC/IEEE Systems and software engineering -- Architecture description. IEEE, Piscataway, NJ, USA 42010:2011
9. Otero, C.E., Peter, A.: Research Directions for Engineering Big Data Analytics Software. IEEE Intell. Syst. 30, 13–19 (2015)
10. NIST: NIST Big Data Interoperability Framework: Volume 1, Definitions. National Institute of Standards and Technology (2015)
11. Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, Sudha: Design Science in Information Systems Research. Management Information Systems Quarterly 28, 75-105 (2004)
12. Jin, X., Wah, B.W., Cheng, X., Wang, Y.: Significance and Challenges of Big Data Research. Big Data Research 2, 59–64 (2015)
13. Mahesh Gudipati, S. Rao, Naju D. Mohan, Naveen Kumar Gajja: Big Data : Testing Approach to Overcome Quality Challenges. Infosys Labs Briefings 11, 65–73 (2013)
14. Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A.S., Buyya, R.: Big Data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing 79-80, 3–15 (2015)
15. Han, R., Lu, X., Xu, J.: On Big Data Benchmarking. In: Zhan, J., Han, R., Weng, C. (eds.) Big Data Benchmarks, Performance Optimization, and Emerging Hardware, vol. 8807, pp. 3–18. Springer International Publishing, Cham (2014)
16. Alexandrov, A., Brücke, C., Markl, V.: Issues in big data testing and benchmarking. In: Sixth International Workshop on Testing Database Systems - DBTest '13, pp. 1–5. ACM Press, New York (2013)
17. Tao, C., Gao, J.: Quality Assurance for Big Data Applications– Issues, Challenges, and Needs. In: The Twenty-Eighth International Conference on Software Engineering and Knowledge Engineering, pp. 375–381. KSI Research Inc. and Knowledge Systems Institute Graduate School, Pittsburgh (2016)
18. Ashoke, S., Haritsa, J.R.: CODD: A dataless approach to big data testing. Proceedings of the VLDB Endowment 8, 2008–2011 (2015)
19. Fan, J., Han, F., Liu, H.: Challenges of Big Data Analysis. National science review 1, 293–314 (2014)

20. Yang, A., Troup, M., Ho, J.W.K.: Scalability and Validation of Big Data Bioinformatics Software. Computational and structural biotechnology journal 15, 379–386 (2017)
21. Alyass, A., Turcotte, M., Meyre, D.: From big data analysis to personalized medicine for all: challenges and opportunities. BMC medical genomics 8, 33 (2015)
22. Kambatla, K., Kollias, G., Kumar, V., Grama, A.: Trends in big data analytics. Journal of Parallel and Distributed Computing 74, 2561–2573 (2014)
23. Barr, E.T., Harman, M., McMinn, P., Shahbaz, M., Yoo, S.: The Oracle Problem in Software Testing: A Survey. IIEEE Trans. Software Eng. 41, 507–525 (2015)
24. Dean, J., Ghemawat, S.: MapReduce. Communications of the ACM 51, 107–113 (2008)
25. Sonnenberg, C., Vom Brocke, J.: Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B. et al. (eds.) Design Science Research in Information Systems. Advances in Theory and Practice, vol. 7286, pp. 381–397. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)