Three Classifications of Big Data-based Software Testing

Pan Liu[*], Jiaqi Yan, and Xiaoyu Song
Faculty of Business Information, Shanghai Business School, Shanghai 201400, China
[*]Corresponding author: liup@sbs.edu.cn

Abstract—**The paper reviews three classifications of big data-based software testing in the last 10 years. Testing for big data system, big data evaluation benchmarks, and the application of big data technology to software testing are studied in the paper. This research helps readers understand the development and classification of big data-based software testing.**

Keywords: big data-based testing; test classification; big data

I. Introduction

In the last decade years, big data has been regarded as a resource [1] to serve human society. For example, in 2009, Google scholars [2] successfully predicted the number of patients with influenza in America based on users' search logs on Internet. Then, big data technology became a research hotspot in the world, gradually being applied to the fields of finance, social interaction, artificial intelligence and software testing. The existing software testing research related to big data technology is divided into three categories, including testing for big data systems, big data evaluation benchmarks, and the application of big data technology to software testing.

II. Testing for big data system

Because big data systems often involve complex data formats, massive data content and complex processing [3], such as task scheduling, data fragmentation, machine fault tolerance, and inter-machine communication in the big data system are all handled by the MapReduce framework. As a result, it is difficult for us to predict which TaskTracker node of the task will be executed in the cluster, nor can we know in advance the location of the node where the map or reduce task is executed [4], so it is difficult to test a big data system. In 2013, Gudipati et al. [5] first proposed that testing big data will become one of the biggest challenges facing the testing industry. They believe that how to develop a testing strategy to deal with structured and unstructured data will become an important research content of software testing. Big data contains a large amount of low-quality, invalid, and redundant data. Storing and processing these data will not only increase the cost of software testing, but also delay the time for software products to market. Using the MapReduce framework to provide a parallel and scalable programming model, data-intensive commercial and scientific research systems can be developed. To test the performance of these big data application systems, Nagdive and Tugnayat [6] reviewed the response time of the system, the maximum number of online users, and the maximum processing capacity of the system. In 2014, Liu et al. [7] believed that traditional performance testing techniques are not suitable for testing big data systems. The performance testing of big data systems needs to be carried out from four aspects: test types, test objectives, test indicators and monitoring indicators. In 2018, Morán et al. [8] proposed a new technique to detect design errors in big data systems. They found possible design flaws in the big data system by simulating different configurations of the equipment. In the experiment, the method proposed by Morán et al. found defects in system design easier than random testing and partition testing. In 2020, Monika et al. [9] presented the testing challenges of big data systems at the International Software Quality Conference (SWQD). They proposed that hardware and storage capabilities are currently a major challenge for testing big data systems.

III. Big data evaluation benchmark

To standardize the testing of big data systems, some big data evaluation benchmarks have been proposed one after another. In 2013, Ghazal [10] et al. proposed an end-to-end big data test benchmark BigBench. The test benchmark covers data models and synthetic data generators, which can solve the test problems of big data systems for structured, semi-structured and unstructured data. In 2014, Zhan et al. [11] developed a scalable and unified big data and AI benchmark test suite BigDataBench. The test benchmark suite not only covers a wide range of application scenarios, but also includes a variety of representative data sets. In 2016, Zhan et al. [12] discussed the conditions that big data evaluation benchmarks need to meet, and elaborated on BigDataBench, the big data benchmark test suite. Then, Jin et al. [13] reviewed the development of big data benchmarks, and Zhou et al. [14] reviewed the current status and trends of big data evaluation benchmarks.

IV. Software testing based on big data technology

Software testing based on big data technology comes from traditional data mining technology. In 2003, Last et al. [15] proposed an input/output-oriented data mining algorithm at the KDD International Academic Conference, and designed redundant test cases for data-driven software systems. In 2005, Van [16] proposed a method of diagnosing business processes by mining event logs, and conducted a consistency test on the software business. In 2007, Menzies et al. [17] predicted software defects by mining the attributes of static code. In their experiments, the average detection

2767-6684/21/$31.00 ©2021 IEEE
DOI 10.1109/DSA52907.2021.00116

probability of predicting factors obtained through data mining methods has reached 71%. In 2010, Hassan et al. [18] mentioned in the future software engineering working group meeting of the international academic conference FSE that mining data in software engineering will be one of the main ways to realize software intelligence. In 2011, Halkidi et al. [19] proposed that data mining can be used to identify and predict software defects and detect software errors. In 2016, Miranskyy et al. [20] mentioned that the software operation log contains the execution path of the software, related events, and user activities. Therefore, big data analysis technology can be used to mine these massive logs to detect defects in the software. Then, Samuelsson [21] used big data technology to analyze software logs and realized software error mining. In 2018, Pan Xing et al. [22] proposed a software quality assurance method based on big data analysis technology. This method is to mine the defects in the software by establishing association rules. In September 2018, we explained the practical application of big data testing technology in software testing in the book "Big Data Testing Technology: data collection, analysis, and test practice" [23]. In 2020, we discovered the hidden danger of hyperlinks on the server by mining the logs on the web server [24].

Acknowledgment

This work was supported in part by the National Social Science Fund General Project of China under Grant 18BTQ058, the National Natural Science Foundation of China under Grant 61502299, and the Business Scholar project in the Shanghai Business School.

References

[1] A. Cuzzocrea, I.-Y. Song, and K. C. Davis, "Analytics over large-scale multidimensional data: the big data revolution!," in Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP, 2011, pp. 101-104.

[2] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," Nature, vol. 457, p. 1012, 2009.

[3] N. Garg, S. Singla, and S. Jangra, "Challenges and Techniques for Testing of Big Data ☆," Procedia Computer Science, vol. 85, pp. 940-948, 2016.

[4] C. Lizhi and Y. Ting, "Challenge and prospect on software test under big data background," Computer Apllications and Software, vol. 31, pp. 5-8, 2014. (in Chinese with English Abstract)

[5] M. Gudipati, S. Rao, N. D. Mohan, and N. K. Gajja, "Big data: Testing approach to overcome quality challenges," Big Data: Challenges and Opportunities, vol. 11, pp. 65-72, 2013.

[6] A. S. Nagdive, M. P. Tembhurkar, and R. Tugnayat, "Overview on performance testing approach in big data," International Journal of Advanced Research in Computer Science, vol. 5, 2014.

[7] Z. Liu, "Research of performance test technology for big data applications," in Information and Automation (ICIA), 2014 IEEE International Conference on, 2014, pp. 53-58.

[8] J. Morán, A. Bertolino, C. de la Riva, and J. Tuya, "Automatic Testing of Design Faults in MapReduce Applications," IEEE Transactions on Reliability, pp. 1-16, 2018.

[9] M. Steidl, R. Breu, and B. Hupfauf, "Challenges in Testing Big Data Systems," in International Conference on Software Quality, 2020, pp. 13-27.

[10] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H.-A. Jacobsen, "BigBench: towards an industry standard benchmark for big data analytics," in Proceedings of the 2013 ACM SIGMOD international conference on Management of data, 2013, pp. 1197-1208.

[11] L. Wang, J. Zhan, C. Luo, Y. Zhu, Q. Yang, Y. He, W. Gao, Z. Jia, Y. Shi, and S. Zhang, "Bigdatabench: A big data benchmark suite from internet services," in High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on, 2014, pp. 488-499.

[12] J. Zhang, W. Gao, L. Wang, et al., "BigDataBench: An Open-source Big Data Benchmark Suite," Chinese Journal of Computers, vol. 39, pp. 196-211, 2016. (in Chinese with English Abstract)

[13] C. Jing, W. Qian, M. Zhou, and A. Zhou, "Benchmarking Data Management Systems: From Traditional Database to Emergent Big Data," Chinese Journal of Computers, vol. 38, pp. 18-34, 2015.

[14] X. Zhou, X. Qin, and Q. Wang, "Big data benchmarks: state-of-art and trends," Journal of Computer Applications, vol. 35, pp. 1137-1142, 2015.

[15] M. Last, M. Friedman, and A. Kandel, "The data mining approach to automated software testing," in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 388-396.

[16] W. M. Van der Aalst, "Business alignment: using process mining as a tool for Delta analysis and conformance testing," Requirements Engineering, vol. 10, pp. 198-211, 2005.

[17] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," IEEE transactions on software engineering, pp. 2-13, 2007.

[18] A. E. Hassan and T. Xie, "Software intelligence: the future of mining software engineering data," in Proceedings of the FSE/SDP workshop on Future of software engineering research, 2010, pp. 161-166.

[19] M. Halkidi, D. Spinellis, G. Tsatsaronis, and M. Vazirgiannis, "Data mining in software engineering," Intelligent Data Analysis, vol. 15, pp. 413-441, 2011.

[20] A. Miranskyy, A. Hamou-Lhadj, E. Cialini, and A. Larsson, "Operational-log analysis for big data systems: Challenges and solutions," IEEE Software, vol. 33, pp. 52-59, 2016.

[21] J. Samuelsson, "Anomaly Detection in ConsoleLogs," ed, 2016.

[22] X. Pan, M. Zhang, and X. Chen, "A Method of Quality Improvement Based on Big Quality Warranty Data Analysis," in 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2018, pp. 643-644.

[23] Liu Pan, "Big Data Testing Technology: data collection, analysis, and test practice," Posts and Telecom Press, 2018. (in Chinese)

[24] P. Liu and W. Huang, "Incremental Data Mining-based Software Failure Detection," International Journal of Performability Engineering, vol. 16, 2020.