Automated Verification of Enterprise Load Tests

Zhen Ming Jiang, Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
Queen's University
Kingston, ON
{zmjiang, ahmed}@cs.queensu.ca

Gilbert Hamann and Parminder Flora
Performance Engineering
Research In Motion (RIM)
Waterloo, ON

ABSTRACT

Many systems ranging from e-commerce websites to telecommunication infrastructures must support concurrent access by hundreds or thousands of users. Many of the field problems are related to these systems not scaling to field workloads instead of feature bugs [1, 11]. To assure the quality of these systems, load testing is a required testing procedure in addition to conventional functional testing procedures, such as unit and integration testing.

Load testing studies the behavior of a system by simulating hundreds or thousands of users performing tasks at the same time. A load test can last from several hours to a few days. Unlike many other software testing mechanisms which focus on testing the system based on a small number (one or two) of users; load testing examines the system's behavior based on concurrent access by a large number (thousands or millions) of users. A typical load test has one or more load generators which simultaneously send requests to the system under test. During the course of a load test, execution logs along with performance data are collected. Execution logs are debug statements that developers insert into the source code. Execution logs record the run time behavior of the application under test. Performance data are collected by resource monitoring tools like PerfMon. Performance data record the resource usage information such as CPU utilization, memory, disk I/O.

Existing load testing research focuses on the automatic generation of load test suites [2, 3, 4, 5, 6, 7, 12]. There is limited work, which proposes the systematic analysis of the results of a load test to uncover potential problems. Unfortunately, looking for problems in a load test is a time-consuming and difficult task due to challenges like: depth and breadth of needed knowledge, time pressure, monitoring overhead, and large volume of generated data. Current practice of load testing analysis is mainly manual and ad-hoc.

We propose an approach which automatically verifies the functional and performance requirements of a load test by mining the readily-available executions logs. Our approach consists of three steps: We first abstract the log lines to execution events [8]. Then, we derive functional models from the events sequences to uncover functional problems [9]. We derive performance models by mining response time distributions to detect performance problems [10]. Our case studies on several open source and large industrial systems show that our approach produces few false alarms and scales well to large dataset.

1. REFERENCES

[1] Applied Performance Management Survey, Oct 2007.

[2] A. Avritzer and B. Larson. Load testing software using deterministic state testing. In *Proceedings of the 1993 ACM SIGSOFT international symposium on Software testing and analysis*, 1993.

[3] A. Avritzer and E. J. Weyuker. Generating test suites for software load testing. In *Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis*, 1994.

[4] A. Avritzer and E. J. Weyuker. The automatic generation of load test suites and the assessment of the resulting software. *IEEE Trans. Softw. Eng.*, 21(9), 1995.

[5] M. S. Bayan and J. W. Cangussu. Automatic stress and load testing for embedded systems. In *30th Annual International Computer Software and Applications Conference*, 2006.

[6] L. C. Briand, Y. Labiche, and M. Shousha. Using genetic algorithms for early schedulability analysis and stress testing in real-time systems. *Genetic Programming and Evolvable Machines*, 7(2), 2006.

[7] V. Garousi, L. C. Briand, and Y. Labiche. Traffic-aware stress testing of distributed systems based on uml models. In *Proceedings of the 28th international conference on Software engineering*, 2006.

[8] Z. M. Jiang, A. E. Hassan, G. Hamann, and P. Flora. An automated approach for abstracting execution logs to execution events. *Journal on Software Maintenance and Evolution: Research and Practice*, 20(4), 2008.

[9] Z. M. Jiang, A. E. Hassan, G. Hamann, and P. Flora. Automatic identification of load testing problems. In *In Proceedings of the 24th IEEE International Conference on Software Maintenance (ICSM)*, 2008.

[10] Z. M. Jiang, A. E. Hassan, G. Hamann, and P. Flora. Automated performance analysis of load tests. In *In Proceedings of the 25th IEEE International Conference on Software Maintenance (ICSM)*, 2009.

[11] E. J. Weyuker and F. I. Vokolos. Experience with performance testing of software systems: Issues, an approach, and case study. *IEEE Trans. Softw. Eng.*, 26(12), 2000.

[12] J. Zhang and S. C. Cheung. Automated test case generation for the stress testing of multimedia systems. *Softw. Pract. Exper.*, 32(15), 2002.