2nd International Workshop on Load Testing of Large Software Systems (LT 2013)

Co-Chairs: ZhenMing Jiang, York University; Marin Litoiu, York University; Ahmed E. Hassan, Queen's University; Paul Vytas, IBM Canada Software Lab

Theme: Cloud Computing and Infrastructure

Integrated Solution: Software Delivery Platform

Abstract:

Nowadays, many software systems are offered in the cloud to millions of users. Field problems of these systems are often due to their inability to scale to field workloads, rather than feature bugs. To assure the quality of these systems, load testing is a required procedure in addition to the conventional functional testing. Problems in a load test could be caused by the system under test, the load generator or the test environment. This one-day workshop intended to bring together researchers and practitioners to discuss the challenges and opportunities of conducting load testing research on large scale software systems.

Workshop Motivations and Justifications

Many large scale software systems ranging from e-commerce websites to telecommunication infrastructures must support concurrent access from millions of users. Studies show that many field problems are due to systems inability to scale to meet user demands, rather than feature bugs. The failure to scale results in catastrophic failures and unfavorable media coverage (e.g., the meltdown of the Firefox website during new releases and the botched launch of Apple's MobileMe). To ensure the quality of these systems, load testing is a required testing procedure in addition to conventional functional testing procedures, like the unit testing and integration testing. Load testing is gaining more importance, as an increasing number of services are being offered in the cloud to millions of users.

Unlike functional testing (e.g., unit testing or integration testing), which focuses on testing a system based on a small number of users, load testing studies the behavior of a system by simulating thousands or millions of users performing tasks at the same time. A typical load test uses one or more load generators that simultaneously send requests to the system under test. There are three phases in a load test: *Design*, *Execution* and *Analysis*. The *Design* phase defines the load that will be placed on the system during testing based on the test objectives. Unlike functional testing, which has clear pass/fail criteria, a load test can have one or more functional and non-functional objectives (e.g., functionality, performance and reliability). The *Execution* phase sets up the test environment and executes the load test. A load test can last from several hours to a few days, during which gigabytes of system behavior data (execution logs and counters) is generated. Execution logs are generated by output statements that developers insert into the source code. Execution logs record software activities (e.g., "User authentication successful") and errors (e.g., "Fail to retrieve customer profile"). Counters can be functionality-related (e.g., number of passed/failed requests) or performance-related (e.g., resource usage information like CPU utilization, memory, disk I/O and network traffic or response time). Some counters (e.g., CPU and memory counters) are collected periodically by monitoring tools like PerfMon. Some other metrics are collected at the end of the tests (e.g., number of passed or failed requests). The *Analysis* phase analyzes these logs and counters and determines if there are any load-related functional and/or non-functional problems.

Load testing is a difficult task, which requires a great understanding of the system under test, the test environment and the generated system behavior data. First, there are thousands or millions of lines of code and possibly many commercial off-the-shelf components interacting with each other for a large scale software system. In addition, problems in a load test could also be caused by the test environment (e.g., buggy load generator or misconfiguration of the test environment). Finally, the generated execution logs and counters are huge. These logs and counters, which are too big to perform in-depth manual analysis, must be analyzed thoroughly to uncover any load-related problems. Unfortunately, load testing has received relatively little attention in the software testing research community. Industry has been focused primarily on creating tools to automatically drive specified load into the system under test (e.g., HP LoadRunner and Apache JMeter). We believe that the current limited research on load testing is mainly due to two reasons: first, there is limited access to large scale multi-user systems and load testing infrastructures, as many of such systems are developed in-house in commercial settings. Second, scalability is not as big of a concern for most prototype systems developed by researchers. However, as more and more services are offered in the cloud for millions of users, research on the analysis of load testing results has become essential.

Goals and Outcomes

Load testing is an important and practical research topic. Yet, little research is done on this topic. This workshop was intended to raise awareness of the importance of load testing research and facilitate collaborations between the academic and industrial researchers. Our ultimate goal of this workshop was to establish and grow an active community around load testing.

Following the success of previous LT 2012 workshop, which was co-located with the Fifth IEEE International Conference on Software Testing, Verification and Validation (ICST 2012), the attendees of LT 2013 consisted of a mixture of academic researchers and industrial practitioners. The workshop presented achievements and challenges of research on load testing large scale software systems and provided networking opportunities to interact with leading load testing researchers and practitioners.

Workshop Structure and Agenda

LT 2013 consisted of several technical sessions and a panel discussion session. The workshop was designed to be interactive with many discussion slot assigned throughout the schedule. For each technical session, there was a slot on discussing the general topics presented in this session. The panel discussion brought together leading industrial practitioners and world-class academic researchers to discuss the opportunities and challenges of conducting research on load testing large scale software systems.

The workshop included insightful presentations and panel discussions on the following topics:

- Developing realistic load tests;
- Optimizing execution of load tests to reduce the duration of load tests;
- Efficient analysis of system behavior data (execution logs and counters);
- Leveraging cloud computing infrastructure to conduct load testing;
- Leveraging extensive software testing research to solve load testing challenges;
- Case studies and experience reports on load testing large scale software systems.