Performance Testing in Software Development: Getting the Developers on Board

Lubomír Bulej
Charles University, Prague, Czech Republic
lubomir.bulej@d3s.mff.cuni.cz

1. ABSTRACT

Society is increasingly dependent on software-intensive systems that are required to interact with a huge number of users and respond in timely manner. Failing that often results in users and consumers not being able to access the advertised services or products, or worse, citizens unable to access vital services provided by the state. In such systems, failures are more likely to be caused by performance issues than by faulty implementation of some features [3, 1].

Modern development processes for general-purpose software systems typically focus on managing complexity to deliver correctly functioning software on time, and best software development practices frown upon premature optimization. With other aspects of software design and construction put above performance concerns, performance becomes a secondary concern that only needs to be addressed if the system's performance turns out to be unsatisfactory.

This contrasts with real-time systems, where meeting real-time performance requirements is essential, and performance is a primary design concern that permeates the development process and the resulting system as a whole. Consequently, the overall performance of the system is a concern that cannot be addressed locally—it must be designed into the system, and strictly controlled throughout its construction.

Simply adopting the process of real-time system development for the development of general-purpose systems is not possible. The size, complexity, and the depth of the software stack used to build general-purpose software-intensive systems typically dwarfs that of the special-purpose mission- or safety-critical real-time systems. The level of control that can be exerted over individual elements of real-time systems either does not scale, or is not possible at all, in addition to performance requirements being usually much less precise, and not easily expressed in terms of latencies or deadlines.

The productivity of developers during development is also an important aspect. Where real-time systems limit development flexibility to maintain control over performance (by avoiding many features of modern runtime platforms unless their performance impact can be sufficiently controlled), general-purpose systems limit the control over performance to maintain development flexibility (by promoting the use of sophisticated frameworks and advanced runtime platforms to manage complexity and increase productivity).

While this looks like a stalemate, we believe that progress can be made by making performance visible to the developers, similarly to how functional unit testing made defects visible and refactoring possible, ultimately improving software quality. Our experimental study [2] shows that developers only see performance when they consciously decide to investigate it, but often introduce code patterns or code modifications based on performance assumptions that may be incorrect, because they are not based on actual performance observations. By making performance visible through testing, we aim to increase the level of performance awareness—the ability to observe performance and act on these observations—so that it permeates the development process, ultimately improving our ability to design performant software-intensive systems without relying on full control over all performance-relevant aspects of the system.

In this talk, we will discuss some challenges in extending the existing performance testing approaches, especially where the scalability w.r.t. large-scale systems is concerned.

2. BIOGRAPHY

Lubomír Bulej is an assistant professor at the Department of Distributed and Dependable Systems, Charles University, Prague, Czech Republic. His primary research interests include performance-related topics focusing on performance evaluation, testing, and monitoring. In addition to performance, his research interests include also dynamic program analysis, with specific focus on making the programs running on the Java (and Dalvik) Virtual Machines more observable. He holds an MSc. from the Czech Technical University in Prague, and a PhD from the Charles University in Prague. He is a member of the ACM.

3. REFERENCES

[1] J. Dean and L. A. Barroso. The tail at scale. *Commun. ACM*, 56(2):74–80, Feb. 2013.

[2] V. Horký, F. Haas, J. Kotrč, M. Lacina, and P. Tůma. Performance regression unit testing: A case study. In *EPEW'13*, volume 8168 of *LNCS*, pages 149–163. Springer, 2013.

[3] 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):1147–1156, Dec. 2000.

ICPE'16 March 12–18, 2016, Delft, Netherlands
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4147-9/16/03.
DOI: http://dx.doi.org/10.1145/2859889.2880448