# The Ninth International Workshop on Load Testing and Benchmarking of Software Systems (LTB 2021)

Alexander Podelko
MongoDB
Stamford, CT, USA
alex.podelko@mongodb.com

Tse-Hsun (Peter) Chen
Concordia University
Montreal, Quebec, Canada
peterc@encs.concordia.ca

Hamzeh Khazaei
York University
Toronto, Ontario, Canada
hkh@eecs.yorku.ca

## ABSTRACT

The Ninth International Workshop on Load Testing and Benchmarking of Software Systems (LTB 2021) is a full-day virtual event bringing together software testing researchers, practitioners and tool developers to discuss the challenges and opportunities of conducting research on load testing and benchmarking software systems. The workshop, co-located with the 12th International Conference on Performance Engineering (ICPE 2021), is held on April 19th, 2021 in Rennes, France.

## CCS CONCEPTS

• **General and reference → Performance**; • **Software and its engineering → Software performance**.

## KEYWORDS

load testing; benchmarking; software systems; performance

**ACM Reference Format:**
Alexander Podelko, Tse-Hsun (Peter) Chen, and Hamzeh Khazaei. 2021. The Ninth International Workshop on Load Testing and Benchmarking of Software Systems (LTB 2021). In *Proceedings of the 2021 ACM/SPEC International Conference on Performance Engineering (ICPE '21), April 19–23, 2021, Virtual Event, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3427921.3450223

## 1 INTRODUCTION

Software systems (e.g., smartphone apps, desktop applications, telecommunication infrastructures, cloud and enterprise systems, etc.) have strict requirements on software performance. Failure to meet these requirements may cause business losses, customer defection, brand damage and other serious consequences. Hence, in addition to conventional functional testing, the performance of these systems must be verified through load testing or benchmarking to ensure quality service.

Load testing and benchmarking software systems are difficult tasks, which require a great understanding of the system under test and customer behavior. Practitioners face many challenges such as tooling (choosing and implementing the testing tools), environments (software and hardware setup) and time (limited time to design, test, and analyze). This one-day workshop brings together software testing researchers, practitioners and tool developers to discuss the challenges and opportunities of conducting research on load testing and benchmarking software systems.

## 2 THE WORKSHOP

The workshop is held on April 19th, 2021 virtually (originally scheduled in Rennes, France), and consists of two keynotes, five research papers, two industry presentations, and a panel.

### 2.1 Keynotes

**Kishor Trivedi** (Duke University)
**Title:** Accelerated Life-testing Applied to Software Systems
**Abstract:** An important metric of software reliability is the mean-time-to-failure (MTTF) of the software system. To estimate this metric, a straightforward method is first collecting a sufficient number of samples of software inter-failure times and then using this sequence of inter-failure times to statistically infer the estimate of its mean and a confidence interval. However, this process is hindered by the fact that the samples of software inter-failures are time-consuming to collect, especially for highly reliable software systems. Furthermore, large and complex software systems are known to contain a significant number of elusive bugs known as Mandelbugs. One sub-type of Mandelbugs is known as aging-related bugs. Another subtype is known as concurrency bugs. Mandelbugs are triggered not just by the inputs or the workload presented to the software but also by the execution environment of the software, such as the operating system and other concurrently running software. Many different factors in the execution environment affect this type of failure occurrence. Accelerated-life testing (ALT) is a known systematic method that has been extensively applied in speeding up the experimental estimation of MTTFs of high-reliability hardware systems. The application of ALT in the context of software systems is the subject of this talk.
**Petr Tuma** (Charles University)
**Title:** Tracking Performance of the Graal Compiler on Public Benchmarks
**Abstract:** For the past three years, we have used several public Java benchmarks (DaCapo, ScalaBench, Renaissance, SPECjvm2008) to track the performance changes introduced by the daily development changes of the Graal compiler. The talk will outline how we tackle common measurement issues such as measurement scheduling and change detection, summarize observed parameters of the performance changes themselves, and then discuss factors that impact the usefulness of such testing for the compiler development process, for example (1) what changes are useful to report (or not), (2) what changes are missed by the public benchmarks, or (3) what

*ICPE '21, April 19–23, 2021, Virtual Event, France*

ACM ISBN 978-1-4503-8194-9/21/04.
https://doi.org/10.1145/3427921.3450223

aspects of the compiler behavior make such testing difficult (or easy).

## 2.2 Research Papers

Five papers are accepted and published in the proceedings:

(1) Franz Bender, Jan Jonas Brune, Nick Lauritz Keutel, Ilja Behnk, and Lauritz Thamsen. *PIERES: A Playground for Network Interrupt Experiments on Real-Time Embedded Systems in the IOT.*
(2) Wajdi Halabi, Daniel Smith, Linh Ngo, Amy Apon, John Hill, Jason Anderson, and Brandon Posey. *Viability of Azure IoT Hub for Processing High Velocity Large Scale IoT Data.*
(3) Sören Henning and Wilhelm Hasselbring. *How to Measure Scalability of Distributed Stream Processing Engines?*
*(4) George Kousiouris and Dimosthenis Kyriazis.* Enabling Containerized, Parametric and Distributed Database Deployment and Benchmarking as a Service.
(5) Adriano Lange, Marcos Sunyé, and Tiago Kepe. *Performance Interference on Key-Value Stores in Multi-tenant Environments: When Block Size and Write Requests Matter.*

## 2.3 Industrial Talks

**Xiaosong Lou** (BlackLine)
**Title:** Concurrent User Modeling - An Alternative Approach to Classic Queuing Theory
**Abstract:** Concurrent User is one of the key performance metrics in a system. Many performance issues that can only be exposed under load are related to the increased number of concurrent users. Concurrent User is a Random Variable. Traditionally it is represented as state probabilities in classic Queuing Analysis. Due to the complexity and restrictions of these models, there has not been a lot of real applications on the subject of Concurrent Users. We propose an analytical alternative to the classic queuing theory for estimating Concurrent User. This model helps us determine whether the simulated workload is a proper representation of the expected production scenario.

**Andreas Grabner** (Dynatrace)
**Title:** Performance as a Self-Service based on SLIs/SLOs with Keptn
**Abstract:** Inspired by how companies like Paypal, Intuit or Dynatrace have been implementing Performance as a Self-Service we included this use case into Keptn - a CNCF Open Source project. Keptn provides Performance as a Self-Service by automating deployment, testing and evaluation a new artifact (e.g: container). Keptn queries custom defined SLIs (Service Level Indicators) from multiple data sources (testing tools, monitoring tools ...), automatically validates them against SLOs (Service Level Objectives) and provides this feedback through ChatOps, the Keptn API or the Keptns Bridge. Join this session and learn how to setup Keptn, how to define SLIs, SLOs and the tests that should be executed and how to make it available to anybody in your organization as a self-service option.

## 2.4 Panel

The "Performance Testing in DevOps" panel is scheduled to discuss the modern trend and challenges of performance testing and benchmarking and how they get integrated in agile development and DevOps.

# 3 ORGANIZERS

**Alexander Podelko** is a staff performance engineer at MongoDB, responsible for performance testing and optimization of the MongoDB server. Before joining MongoDB, he worked for Oracle/Hyperion, Aetna, and Intel. Alexander talks and writes about performance-related topics, advocating tearing down silo walls between different groups of performance professionals. He currently serves as a director for the Computer Measurement Group (CMG), an organization of performance and capacity professionals. He received his PhD in Computer Science from Gubkin University and his MBA from Bellevue University. More information at
https://www.alexanderpodelko.com/Bio.html

**Tse-Hsun (Peter) Chen** is an Assistant Professor in the Department of Computer Science and Software Engineering at Concordia University, Montreal, Canada. He leads the Software PErformance, Analysis, and Reliability (SPEAR) Lab, which focuses on conducting research on performance engineering, program analysis, log analysis, production debugging, and mining software repositories. His work has been published in flagship conferences and journals such as ICSE, FSE, TSE, EMSE, and MSR. He serves regularly as a program committee member of international conferences in the field of software engineering, such as ASE, ICSME, SANER, and ICPC, and he is a regular reviewer for software engineering journals such as JSS, EMSE, and TSE. Dr. Chen obtained his BSc from the University of British Columbia, and MSc and PhD from Queen's University. Besides his academic career, Dr. Chen also worked as a software performance engineer at BlackBerry for over four years. Early tools developed by Dr. Chen were integrated into industrial practice for ensuring the quality of large-scale enterprise systems. More information at: http://petertsehsun.github.io.

**Hamzeh Khazaei** is an assistant professor of Computer Science in the Department of Electrical Engineering and and Computer Science at York University. Previously he was an assistant professor at the University of Alberta, research associate at the University of Toronto and a research scientist at IBM, respectively. He received his PhD degree in Computer Science from the University of Manitoba where he extended queuing theory and stochastic processes to accurately model the performance and availability of cloud computing systems. More information at https://www.eecs.yorku.ca/~hkh.

# 4 PROGRAM COMMITTEE

Eduardo Cunha de Almeida, UFPR, Brazil; Zhen Ming (Jack) Jiang, York University, Canada; Johannes Kroß, Fortiss, Germany; Gerson Sunyé, University of Nantes, France; Heng Li, Polytechnique Montréal, Canada; Jinfu Chen, Huawei, Canada; Weiyi Shang, Concordia University, Canada;

# 5 STEERING COMMITTEE MEMBERS

Ahmed E. Hassan, Queen's University, Canada; Marin Litoiu, York University, Canada; Zhen Ming (Jack) Jiang, York University, Canada.