

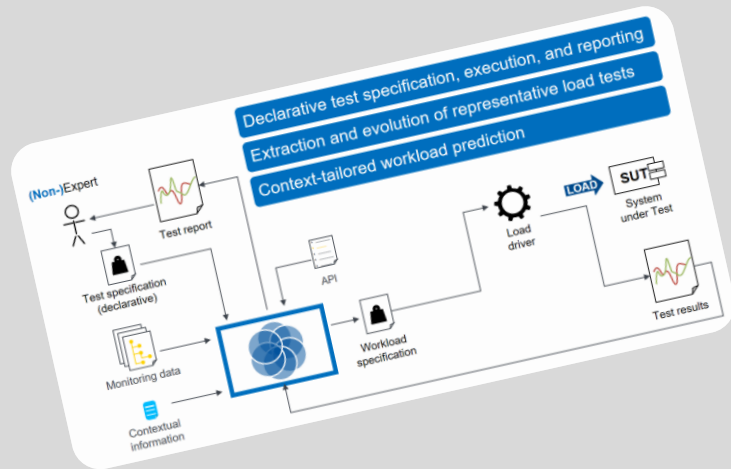
Test-based Scalability and Resilience Assessment of Microservice-based Software Systems

André van Hoorn



October 28, 2019, Berlin Germany

International Workshop on Governing Adaptive and Unplanned Systems of Systems (GAUSS) @ ISSRE 2019



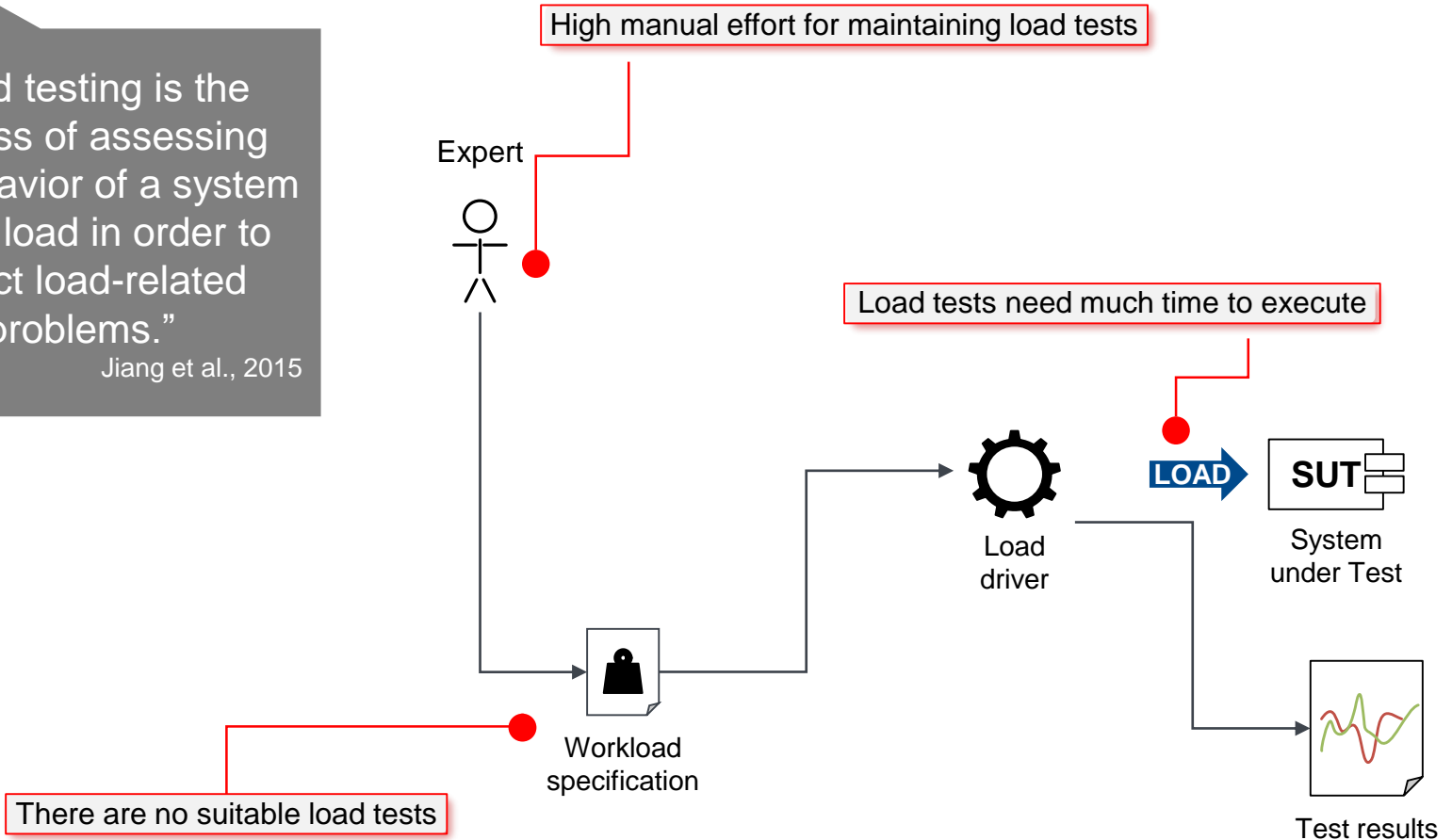
University of Stuttgart



Load Testing – Approach and Challenges

“Load testing is the process of assessing the behavior of a system under load in order to detect load-related problems.”

Jiang et al., 2015

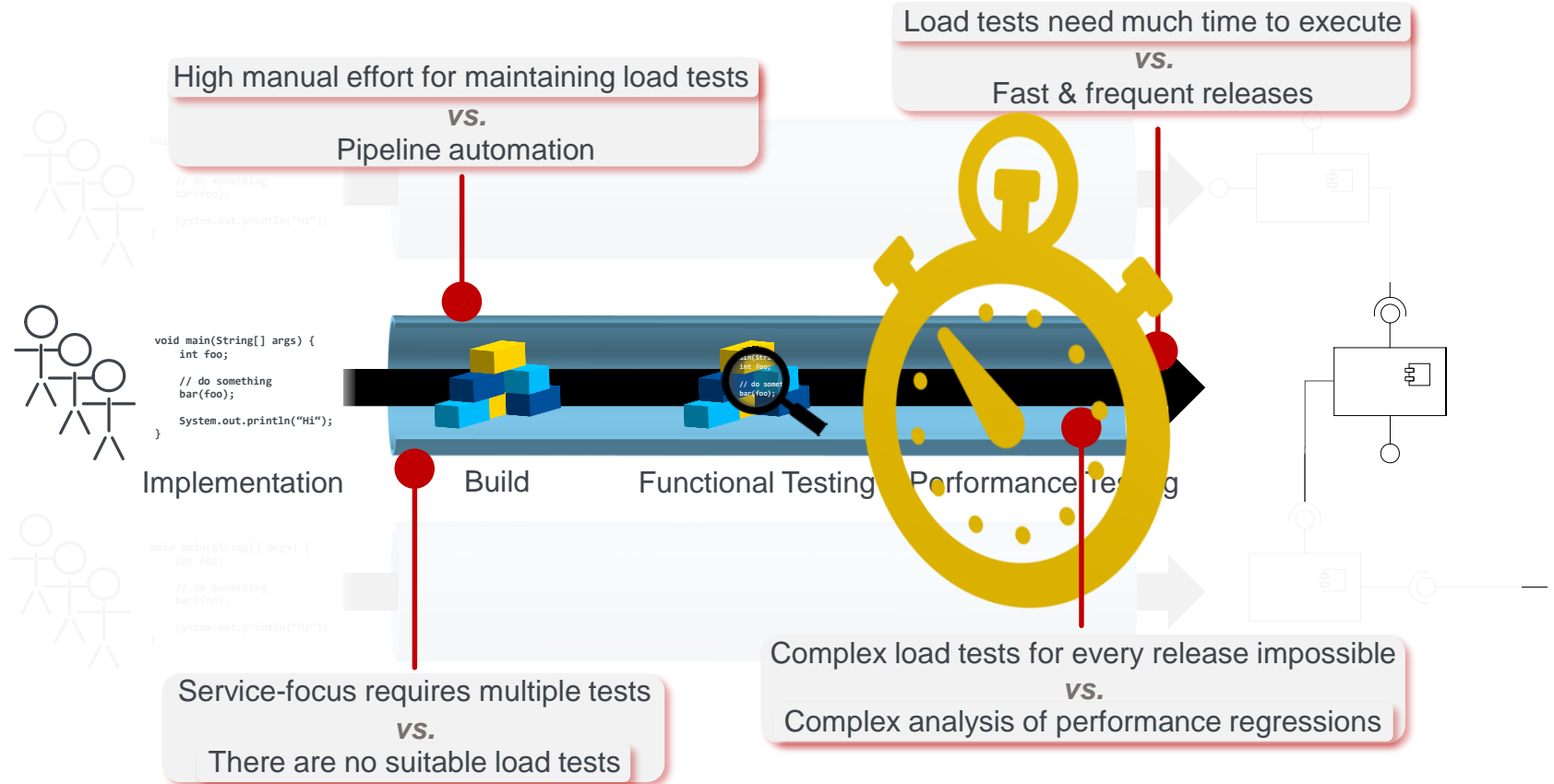


Bezemer et al., ICPE 2019

Schulz et al., ICPE 2018

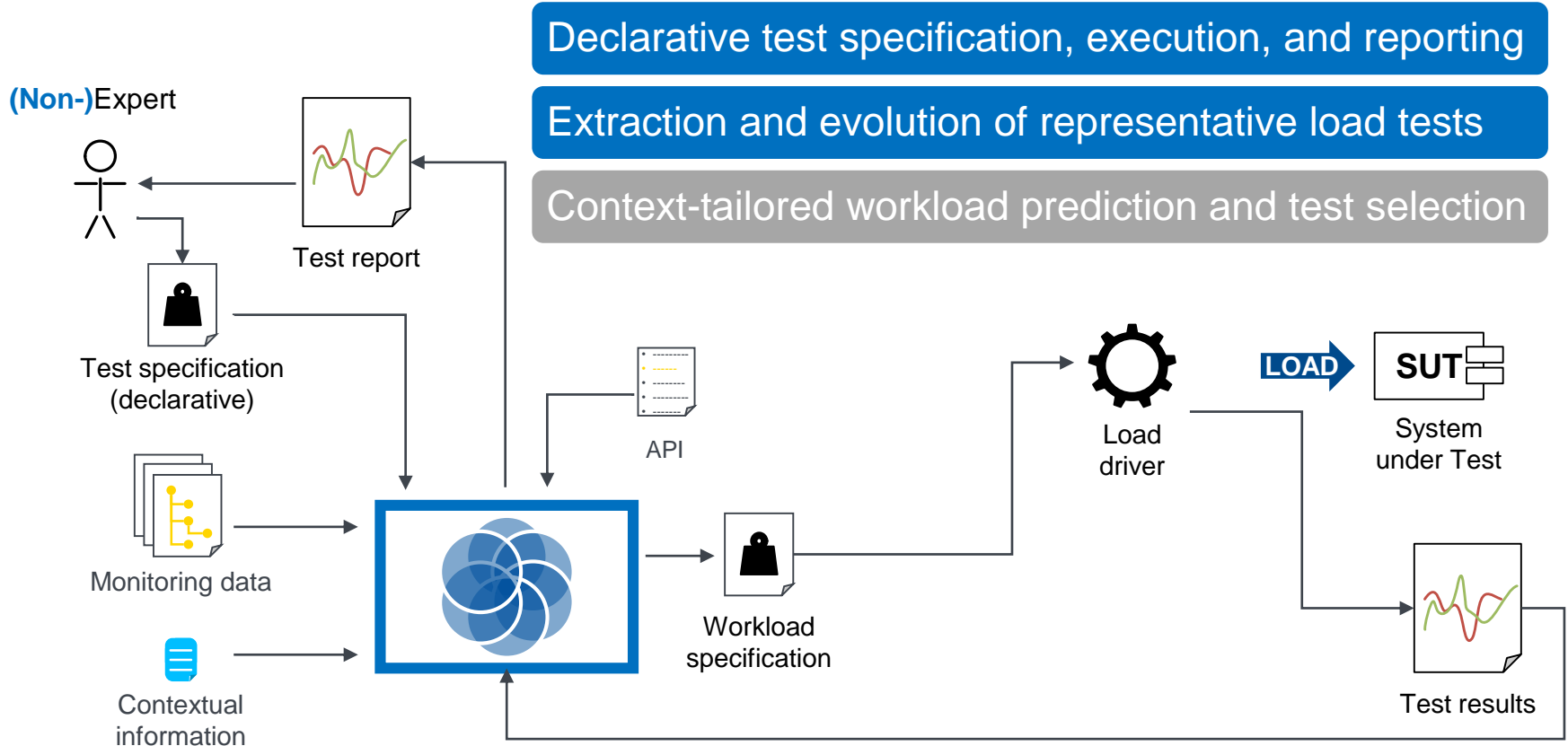
Load Testing of Microservices in Continuous Delivery Pipelines

... How Problems Get Worse



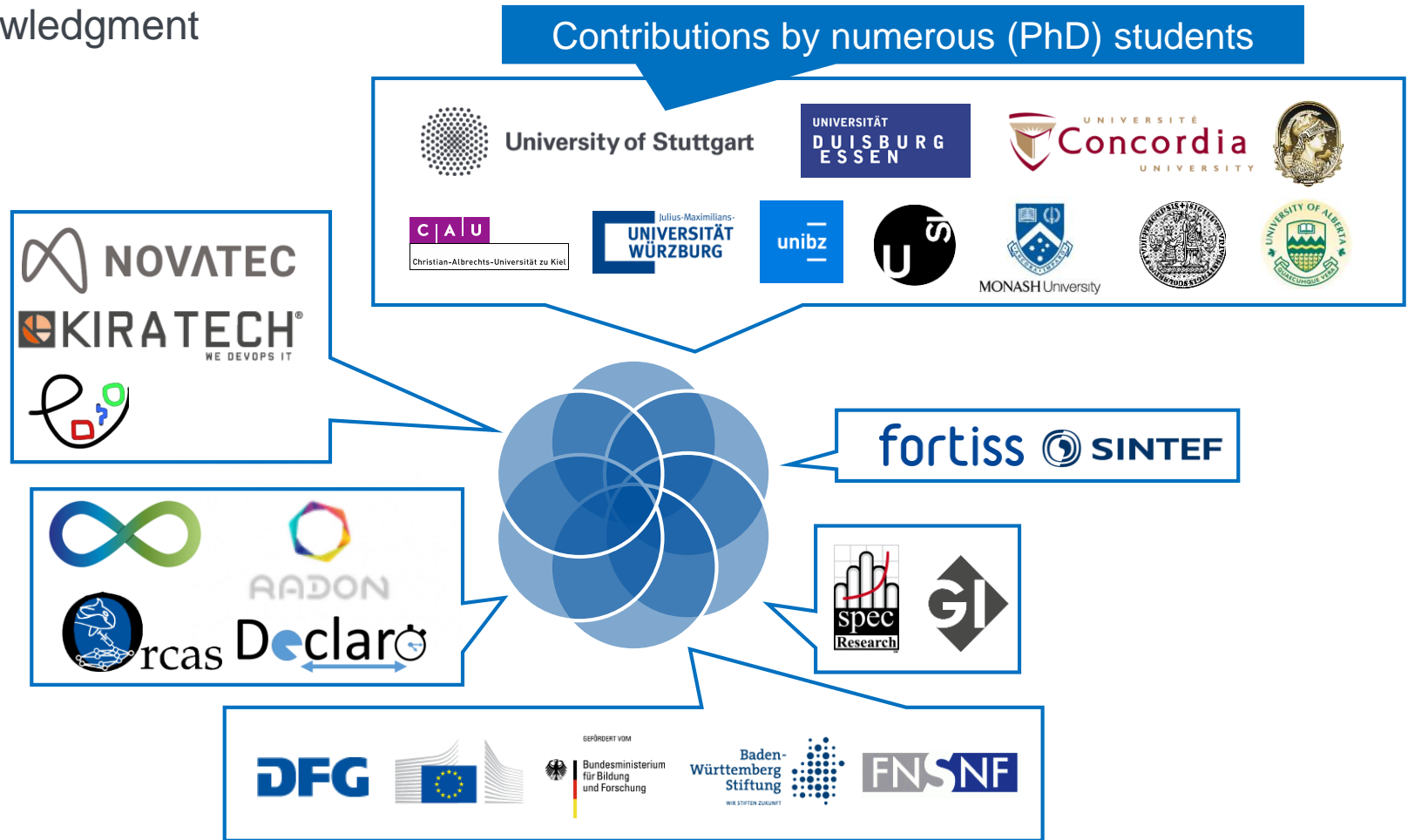
Our Approach

Overview of our Approach



This Work is a Joint Effort!

Acknowledgment




Declarative test specification, execution, and reporting

Extraction and evolution of representative load tests

Context-tailored workload prediction and test selection

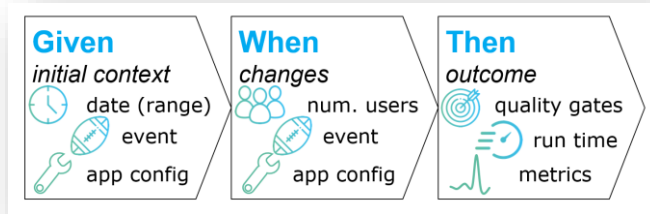
Behavior-Driven Load Testing

Adopting idea from Behavior-Driven Development (BDD)

Schulz et al., ICPE 2019 

For functional testing

- BDLT language – structure and example

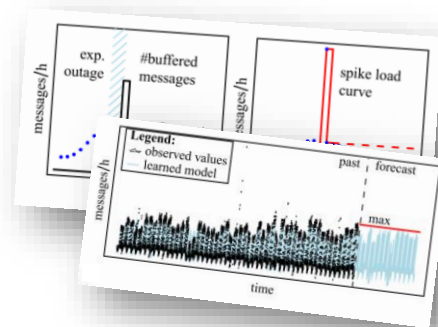


Given the next Black Friday,
when varying the CPU cores between 1 and 4,
then run the experiment for 1h
and ensure the maximum CPU utilization is less than 60%.

- Mapping to Continuity and BenchFlow for load test generation

- Evaluation:

Expressiveness? Use?
Benefits and Limitations?



Formulated industrial load test scenarios, using operational data

Expert interviews

“NL helps understanding and communication (with non-experts)”

“Could replace manual tests”

“Some concerns hard to express”

Concern-driven Analysis and Reporting

Okanovic et al., ICPE 2019



Addressed problem:

- Exact concerns not made explicit before performance evaluation
- Performance evaluation reports not appropriate (e.g., not tailored to concern, results not explained)

What was the **average latency** of the system, **5 seconds** after the experiment start

minimum average maximum

latency number of active users
connection time response time traffic
number of successful requests

0 1 2 3 4 5 6 7 8 9 10
20 30 40 50 60 70 80 90
100 200 300 400 500 600 700
800 900 1000

milliseconds seconds minutes hours
days

Template-based NL specification of concerns

- Performance evaluation (configuration, execution)
- Data filtering
- Report generation

The average latency of the system was **140.35ms**, 5 seconds after the experiment start.

Experiment Result

Configuration and Analysis

The configured query did trigger a **loadtest**. The chosen loadtest tool was **JMeter**. JMeter performed the loadtest on the domain **www.example.com/test** with a load of **1000 users**. Each user sent exactly **1 request**. The evaluation started at **24.10 8:55:51** and ended at **24.10 8:56:51**. JMeter collected the following metrics:

- Latency
- number of active Threads (Users)
- Connection Time
- Successful Request
- Traffic

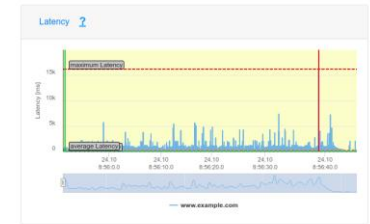
The inspected metrics were recorded over the course of **1 minute and 0 seconds and 916 milliseconds**. During this time **14156 requests** were saved to the analysis result.

Query

What was the maximum latency of the system when the number of active users was = 1000?

The maximum latency of the service was **376ms**, when the number of active users was = 1000.

Latency




Latency (Definition): Latency is the amount of time a message takes to traverse a system. In a computer network, it is an expression of how much time it takes for a packet of data to get from one designated point to another. It is measured as the time required for a request to be sent to the server and returned to its sender. Latency depends on the speed of the transmission medium and the delays in the transmission by devices along the way. A low latency indicates a high network efficiency.

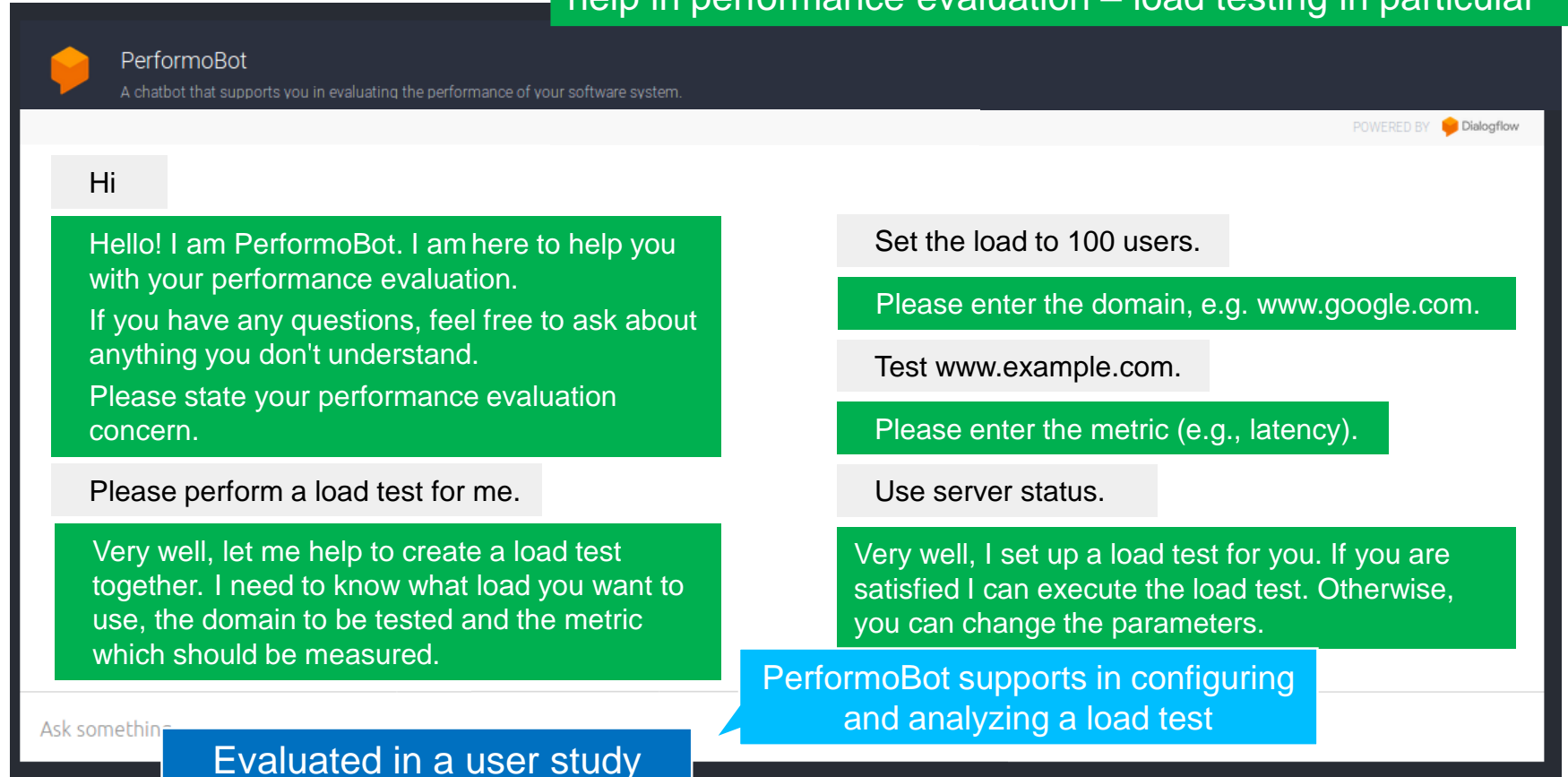
The **minimum Latency** was 0ms at **24.10 8:55:52.309**.
The **average Latency** was **193.78ms**.
The **maximum Latency** was **16172ms** at **24.10 8:56:38.904**.

Tailored report with NL explanations


Concern-driven Analysis and Reporting – Chatbot Extension

Okanović et al., under review 

Hypothesis: Chatbots used in various domains and could help in performance evaluation – load testing in particular



PerformoBot
A chatbot that supports you in evaluating the performance of your software system.

POWERED BY  Dialogflow

Hi

Hello! I am PerformoBot. I am here to help you with your performance evaluation. If you have any questions, feel free to ask about anything you don't understand. Please state your performance evaluation concern.

Please perform a load test for me.

Very well, let me help to create a load test together. I need to know what load you want to use, the domain to be tested and the metric which should be measured.

Set the load to 100 users.

Please enter the domain, e.g. www.google.com.

Test www.example.com.

Please enter the metric (e.g., latency).

Use server status.

Very well, I set up a load test for you. If you are satisfied I can execute the load test. Otherwise, you can change the parameters.

Ask something

Evaluated in a user study with 47 participants

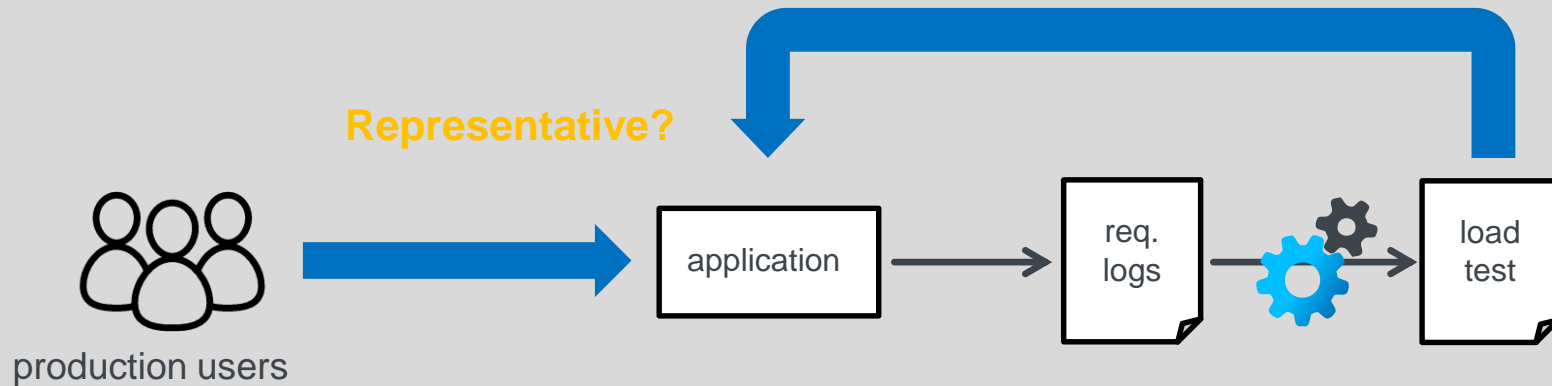
PerformoBot supports in configuring and analyzing a load test

Most helpful to non-experts – incl. education

Declarative test specification, execution, and reporting

Extraction and evolution of representative load tests

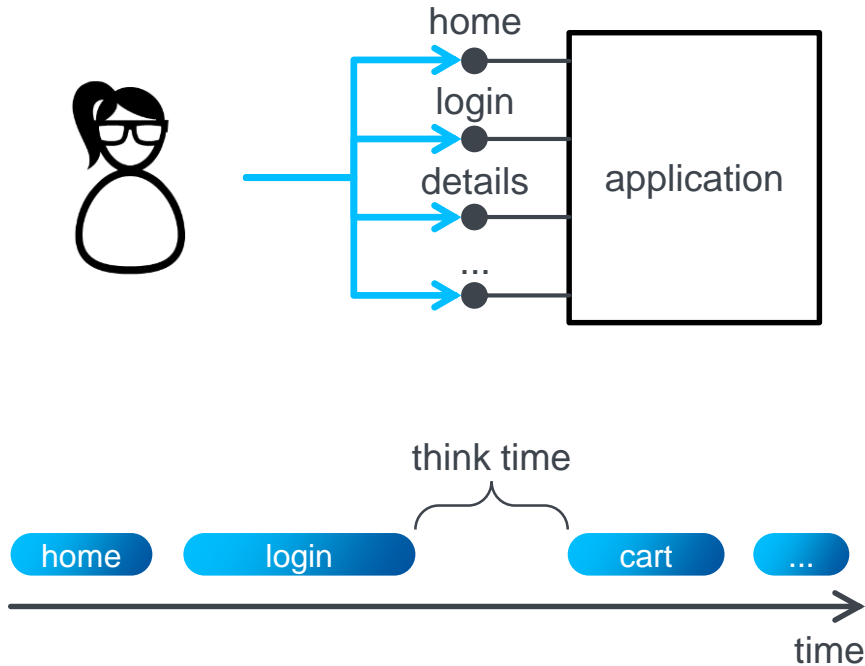
Context-tailored workload prediction and test selection



Background: Session-based Workload

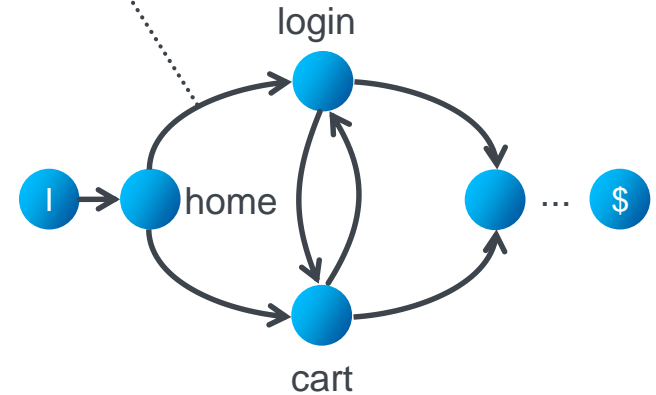
Vögele et al., SoSyM 2018

Schulz et al., MASCOTS 2019



aggregation to Markov chains

probability: 0.7
think time: $\mathcal{N}(10, 5)$

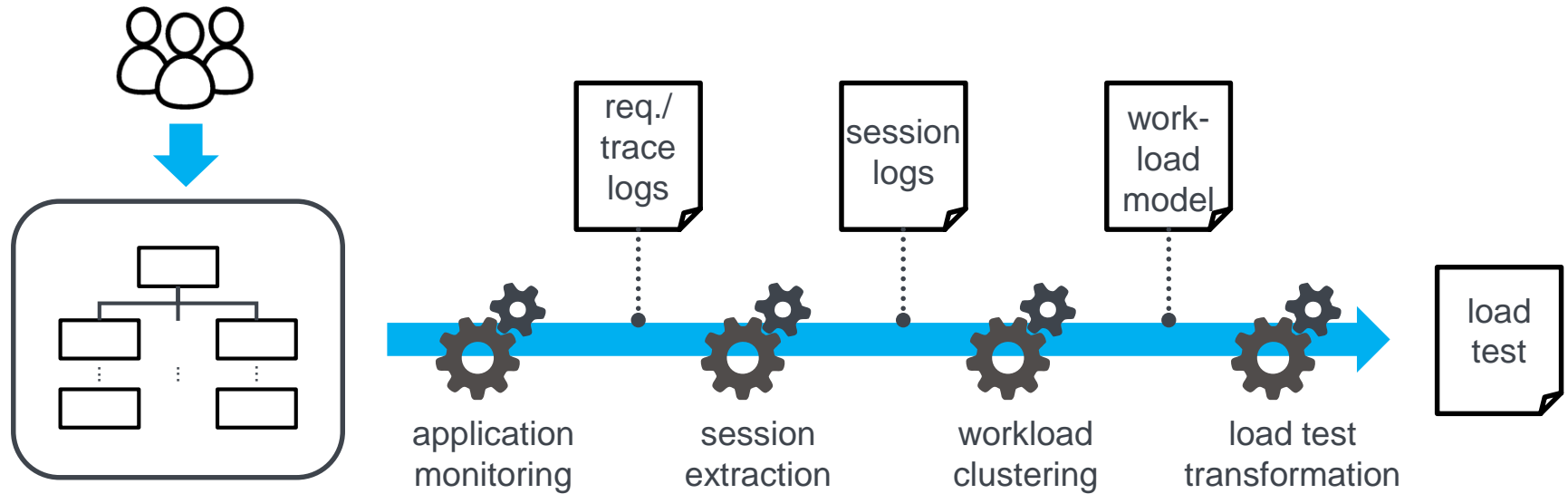


Extraction of Session-based Workload Models and Load Tests


Vögele et al., SoSyM 2018



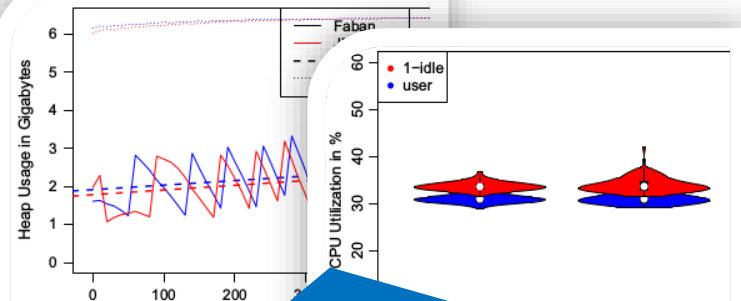
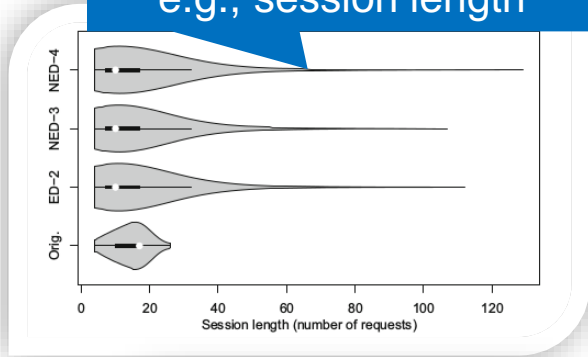
Schulz et al., MASCOTS 2019



Evaluation of the Representativeness

Vögele et al., SoSyM 2018 

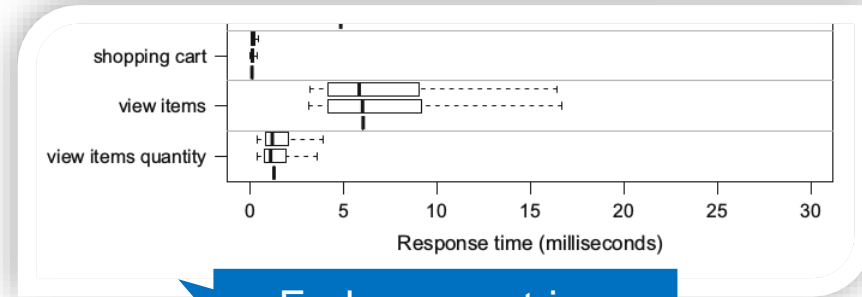
Session-based metrics,
e.g., session length



System-level metrics,
e.g., CPU and memory

Request-based metrics,
e.g., request mix

Request	Orig.	ED-2	NED-3	NED-4	Rel.
<i>(a) Absolute and relative (Rel.) counts (JMeter)</i>					
1. Add to cart	20,625	21,474	21,129	21,217	0.07
2. Cancel order	191	198	176	168	0.00
3. Clear cart	1932	2129	2011	1976	0.01
4. Defer order	2236	2228	2218	2312	0.01
5. Home	19,371	20,119	20,358	20,299	0.07

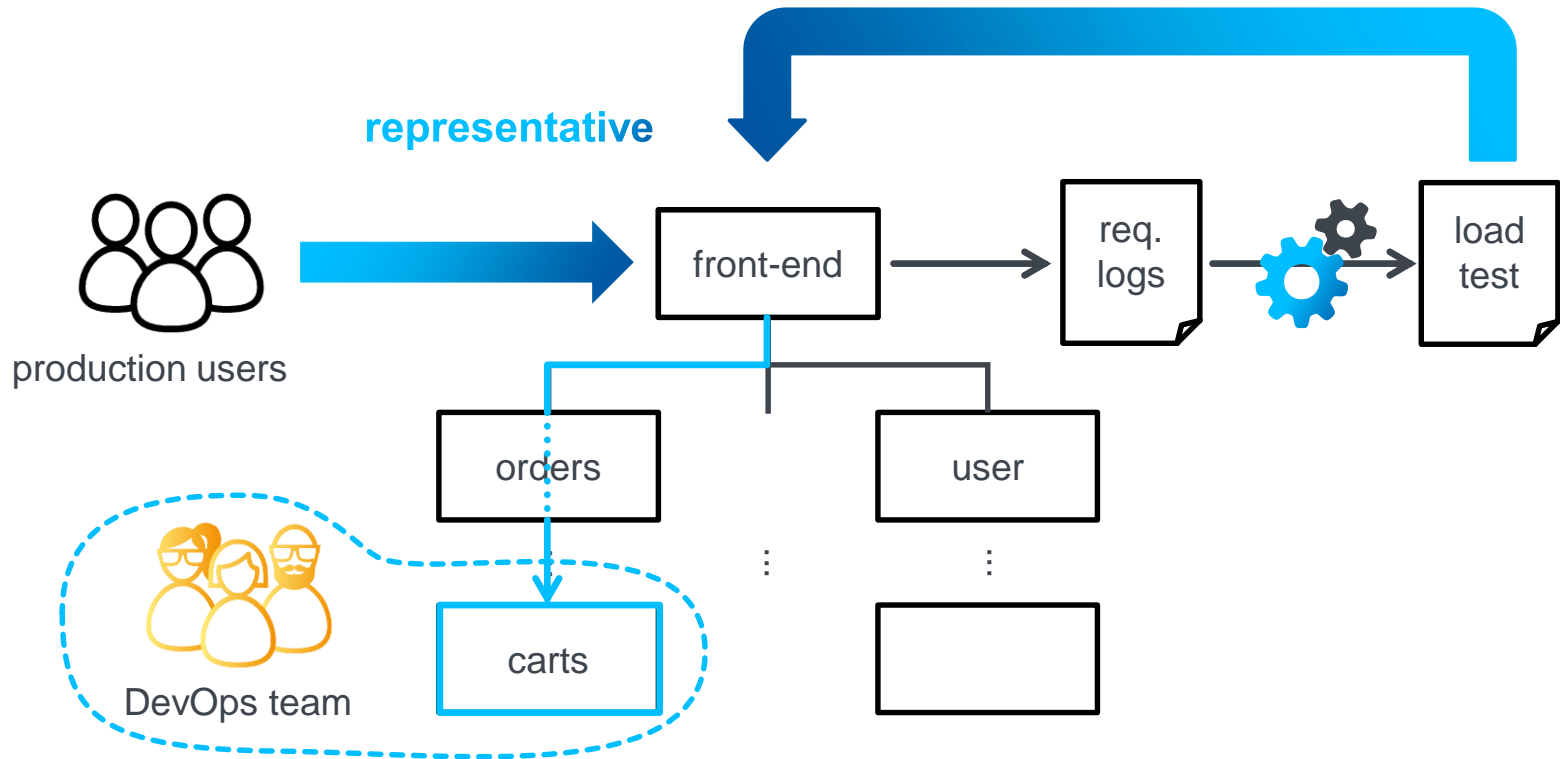


End-user metrics,
e.g., response times


Not considered in this study:
varying workload intensities

Representative Load Testing Lacks in Support for Microservices

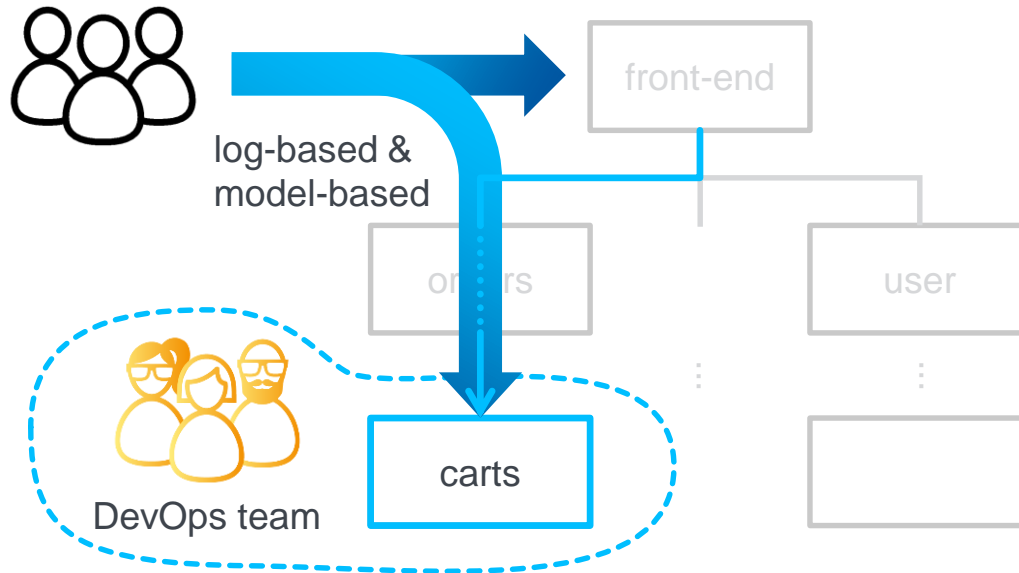
Schulz et al., MASCOTS 2019 



Microservice-tailored Generation of Session-based Workload Models

Schulz et al., MASCOTS 2019 

Idea: Derive representative load tests tailored to individual microservices from the global application workload

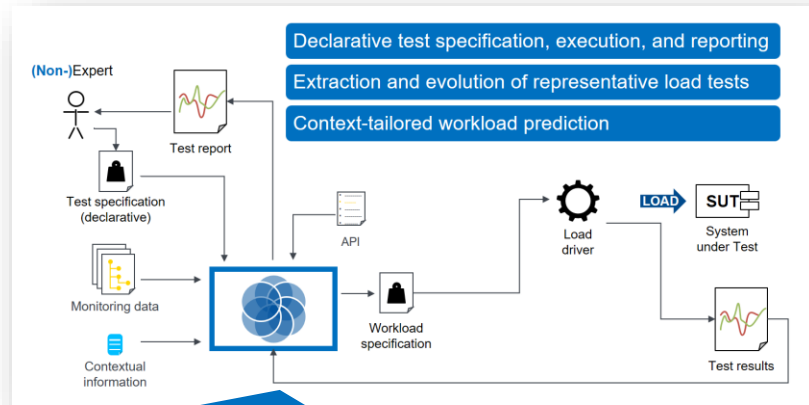


- ✓ less resources
- (✓) shorter execution
- ✓ representative
- ✓ DevOps ready

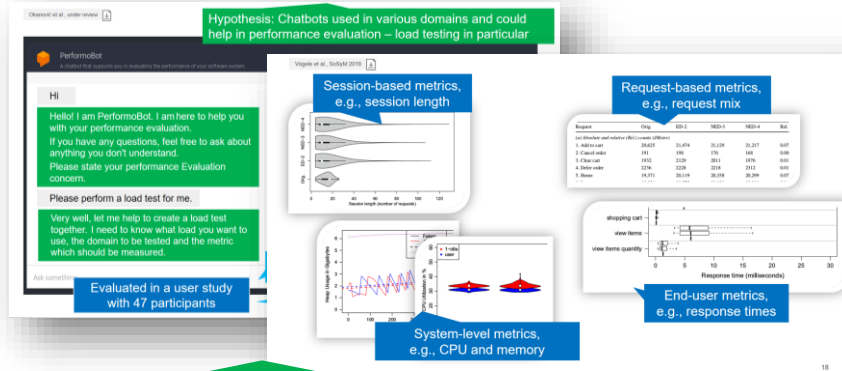
“Load testing is the process of assessing the behavior of a system under load in order to detect load-related problems.”

Jiang et al., 2015

Load testing is challenging – and even more in the context of microservices and DevOps



Overview of our collaborative approach



Selected results:

- declarative specification and reporting
- extraction and evolution of representative load tests

Additional pointers

- Efficient resilience testing van Hoorn et al., ISSRE 2018
- Domain-based scalability testing Avritzer et al., ECSA 2018
- Scalability testing in Scrum Brataas et al., Software 2019
- Automated load test evolution Schulz et al., STVR 2019

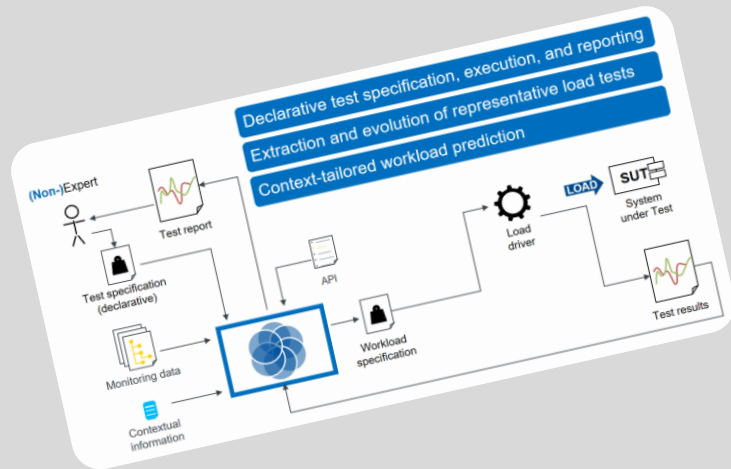
Test-based Scalability and Resilience Assessment of Microservice-based Software Systems

André van Hoorn



October 28, 2019, Berlin Germany

International Workshop on Governing Adaptive and Unplanned Systems of Systems (GAUSS) @ ISSRE 2019



University of Stuttgart



References

Schulz et al., STVR 2019

Henning Schulz, André van Hoorn, Alexander Wert: Reducing the Maintenance Effort for Parameterization of Representative Load Tests Using Annotations. *Software Testing, Verification & Reliability* (2019)

Schulz et al., MASCOTS 2019

Henning Schulz, Tobias Angerstein, Dušan Okanović, and André van Hoorn. Microservice-tailored generation of session-based workload models for representative load testing. *MASCOTS 2019*

Brataas et al., *Software* 2019

Gunnar Brataas, Geir Kjetil Hanssen, Nikolas Herbst, and André van Hoorn. Agile scalability engineering: The ScrumScale method. *IEEE Software* (2019)

Okanovic et al., ICPE 2019

Dusan Okanovic, André van Hoorn, Christoph Zorn, Fabian Beck, Vincenzo Ferme, Jürgen Walter: Concern-driven Reporting of Software Performance Analysis Results. *ICPE Companion 2019: 1-4*

Avritzer et al., ICPE 2019

Alberto Avritzer, Daniel S. Menasché, Vilc Rufino, Barbara Russo, Andrea Janes, Vincenzo Ferme, André van Hoorn, Henning Schulz: PPTAM: Production and Performance Testing Based Application Monitoring. *ICPE Companion 2019: 39-40*

Bezemer et al., ICPE 2019

Cor-Paul Bezemer, Simon Eismann, Vincenzo Ferme, Johannes Grohmann, Robert Heinrich, Pooyan Jamshidi, Weiyi Shang, André van Hoorn, Mónica Villavicencio, Jürgen Walter, Felix Willnecker: How is Performance Addressed in DevOps? A Survey on Industrial Practices. *ICPE 2019: 45-50*

References

Schulz et al., ICPE 2019

Henning Schulz, Dusan Okanovic, André van Hoorn, Vincenzo Ferme, Cesare Pautasso: Behavior-driven Load Testing Using Contextual Knowledge - Approach and Experiences. ICPE 2019: 265-272

Trubiani et al., IST 2018

Catia Trubiani, Alexander Bran, André van Hoorn, Alberto Avritzer, Holger Knoche: Exploiting load testing and profiling for Performance Antipattern Detection. Information & Software Technology 95: 329-345 (2018)

Pitakrat et al., JSS 2018

Teerat Pitakrat, Dusan Okanovic, André van Hoorn, Lars Grunske: Hora: Architecture-aware online failure prediction. Journal of Systems and Software 137: 669-685 (2018)

Vögele et al., SoSyM 2018

Christian Vögele, André van Hoorn, Eike Schulz, Wilhelm Hasselbring, Helmut Krcmar: WESSBAS: extraction of probabilistic workload specifications for load testing and performance prediction - a model-driven approach for session-based application systems. Software and Systems Modeling 17(2): 443-477 (2018)

Avritzer et al., ECSA 2018

Alberto Avritzer, Vincenzo Ferme, Andrea Janes, Barbara Russo, Henning Schulz, André van Hoorn: A Quantitative Approach for the Assessment of Microservice Architecture Deployment Alternatives by Automated Performance Testing. ECSA 2018: 159-174

van Hoorn et al., ISSRE 2018

André van Hoorn, Aldeida Aleti, Thomas F. Düllmann, Teerat Pitakrat: ORCAS: Efficient Resilience Benchmarking of Microservice Architectures. ISSRE Workshops 2018: 146-147

References

Schulz et al., ICPE 2018

Henning Schulz, Tobias Angerstein, André van Hoorn: Towards Automating Representative Load Testing in Continuous Software Engineering. ICPE Companion 2018: 123-126

van Hoorn et al., DATACOM 2018

André van Hoorn and Stefan Siegl. Application Performance Management: Measuring and Optimizing the Digital Customer Experience (e-Book). SIGS DATACOM GmbH (2018)

Düllmann et al., ICPE 2017

Thomas F. Düllmann, André van Hoorn: Model-driven Generation of Microservice Architectures for Benchmarking Performance and Resilience Engineering Approaches. ICPE Companion 2017: 171-172

Heinrich et al., ICPE 2017

Robert Heinrich, André van Hoorn, Holger Knoche, Fei Li, Lucy Ellen Lwakatare, Claus Pahl, Stefan Schulte, Johannes Wettinger: Performance Engineering for Microservices: Research Challenges and Directions. ICPE Companion 2017: 223-226

Heger et al., ICPE 2017

Christoph Heger, André van Hoorn, Mario Mann, Dusan Okanovic: Application Performance Management: State of the Art and Challenges for the Future. ICPE 2017: 429-432

Walter et al., ICPE 2016

Jürgen Walter, André van Hoorn, Heiko Koziolk, Dusan Okanovic, Samuel Kounev: Asking "What"?, Automating the "How"?: The Vision of Declarative Performance Engineering. ICPE 2016: 91-94

van Hoorn et al., ICPE 2012

André van Hoorn, Jan Waller, Wilhelm Hasselbring: Kieker: a framework for application performance monitoring and dynamic software analysis. ICPE 2012: 247-248

References (submitted papers)

Okanović et al., <under review>

Dušan Okanović, Samuel Beck, Lasse Merz, Christoph Zorn, Leonel Merino, André van Hoorn, Fabian Beck: Can a Chatbot Support Software Engineers with Load Testing? Approach and Experiences. *Under review*