Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

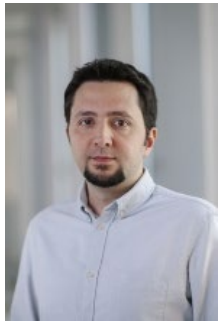
Descartes research

Microservices: A Performance Tester's Dream or Nightmare?

Simon Eismann
University of Würzburg
@simon_eismann

Cor-Paul Bezemer
University of Alberta
@corpaul

Weiyi Shang
Concordia University
@swy351



Dušan Okanović
University of Stuttgart
@okanovic_d

André van Hoorn
University of Stuttgart
@andrevanhoorn

spec

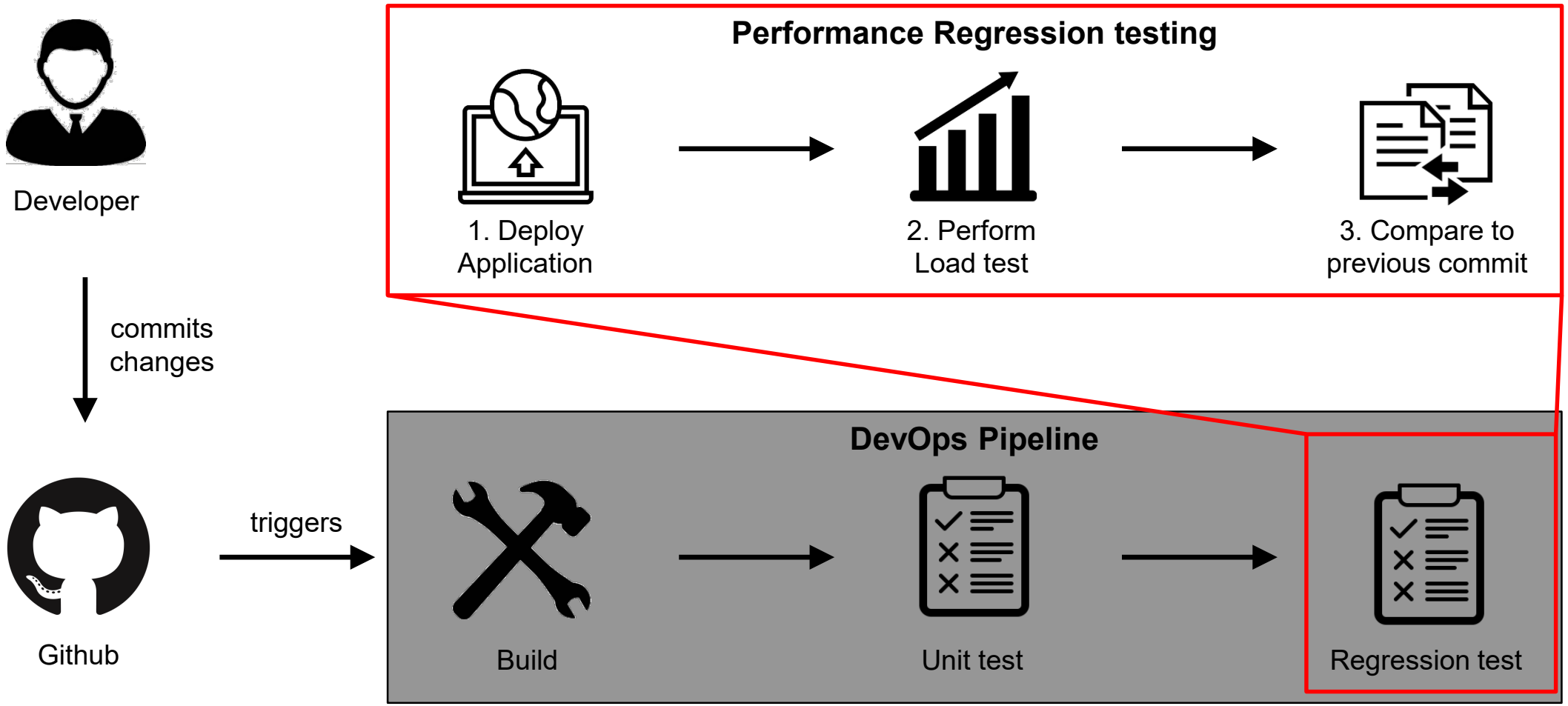
spec RG DevOps Performance

https://research.spec.org/working-groups/rg-devops-performance.html

Artifacts Available acm

Artifacts Evaluated acm Reusable

What is Performance Regression Testing?



Requirements for Performance Testing

R1 A stable testing environment which is representative of the production environment

R2 A representative operational profile (including workload characteristics and system state) for the performance test

R3 Access to all components of the system

R4 Easy access to stable performance metrics

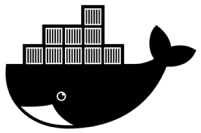
R5 Sufficient time

Microservice traits

T1 Self-containment

T2 Loosely coupled, platform-independent interfaces

T3 Independent development, build, and deployment.



T4 Containers and Container Orchestration

T5 Cloud-native

Microservices - A Performance Testers Dream?

Benefit 1: Containerization

- Containers package environment
- Simplifies setup of test environment

Benefit 2: Granularity

- Individually testable services
- Dependencies via HTTP calls
- Dependencies easily mocked

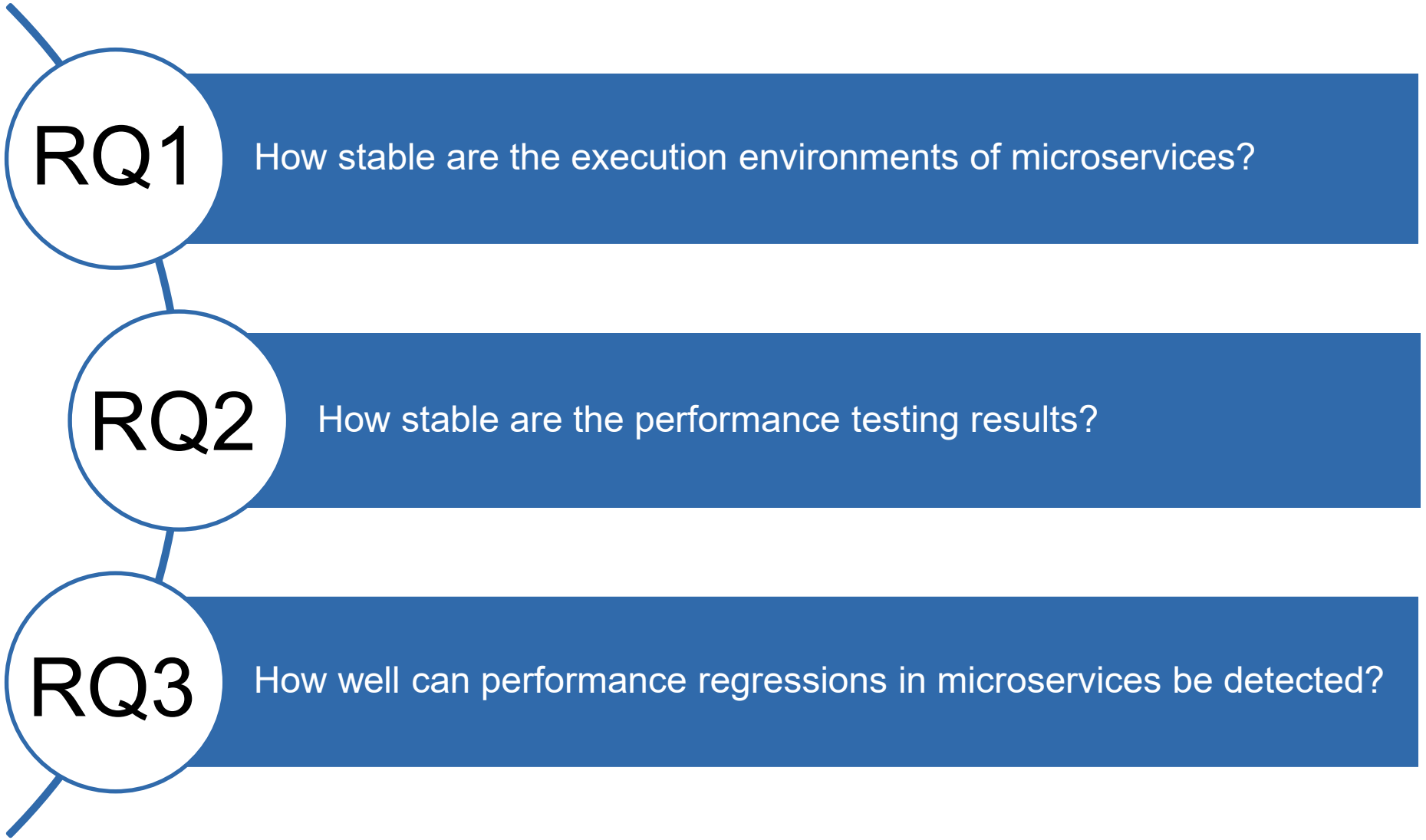
Benefit 3: Easy access to metrics

- Orchestration frameworks simplify metric collection
- Application-level metrics common

Benefit 4: Integration with DevOps

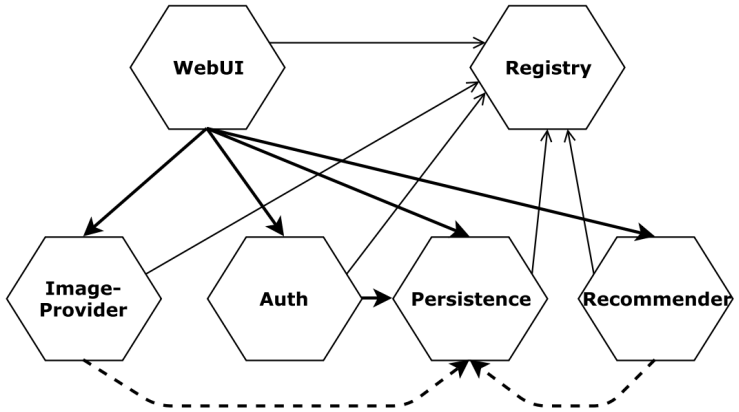
- Size reduces performance test duration
- Performance testing within pipeline

Too good to be true? – Let's test it!



Case Study

TeaStore Benchmarking Application



Scenarios

Scenario	#Nodes	Cores/Node	Memory/Node
Default	20	1	6.5 GB
Balanced	20	1	6.5 GB
LargeVMs	5	4	26 GB
Autoscaling	5	4	26 GB
Regression (baseline)	5	4	26 GB
Regression	5	4	26 GB

Table 1: Cluster size in the different scenarios.

Deployment Platform

docker kubernetes HashiCorp Terraform Google Cloud Platform

Research Question 1 – Selected Findings

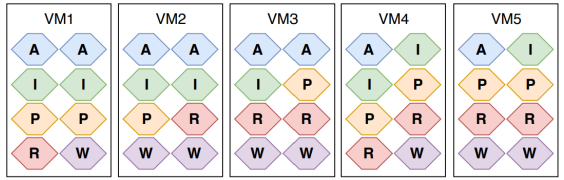
How stable are the execution environments of microservices across repeated runs of the experiments?

Finding 1: The non-deterministic behaviour of the autoscaler results in different numbers of provisioned microservice instances when scaling the same load

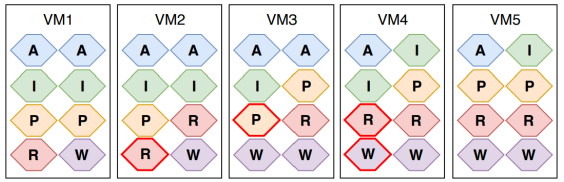
Load	Service	Experiment run 1	2	3	4	5	6	7	8	9	10
700	Auth	4	5	4	4	4	7	4	3	4	3
	WebUI	8	8	8	8	8	8	8	8	8	8
	Recom.	2	2	1	1	1	1	1	1	1	1
	Persist.	8	8	7	6	7	5	6	6	6	6
	Image	4	4	4	4	4	5	3	3	4	4
800	Auth	5	6	4	4	4	4	4	4	4	4
	WebUI	8	8	8	8	8	8	8	8	8	8
	Recom	1	3	1	1	1	2	1	1	1	1
	Persist.	7	8	7	7	7	7	7	7	7	7
	Image	4	5	4	5	4	4	3	4	4	4
900	Auth	5	5	5	5	5	5	4	5	5	3
	WebUI	8	8	8	8	8	8	8	8	8	8
	Recom.	2	2	2	2	2	2	2	2	2	2
	Persist.	8	8	8	8	8	7	7	8	8	7
	Image	5	5	5	5	5	5	5	4	5	4

Table 2: Number of provisioned service instances after twenty minutes of warmup across ten experiment repetitions in the Autoscaling scenario.

Finding 2: Even when fixing the number of provisioned instances of a microservices, their deployment across VMs differs.



(a) Deployment during experiment run 4/10



(b) Deployment during experiment run 5/10

Figure 3: Deployment from two repetitions of LargeVMs scenario with 700 requests/second. The differences in deployment are indicated by thick red borders (A = authentication service, I = ImageProvider service, P = Persistence service, R = Recommender service, W = WebUI service).

Research Question 2 – Selected Findings

How stable are the performance testing results across repeated runs of the experiments?

Finding 1: There exist statistically significant differences between the performance testing results from different scenarios

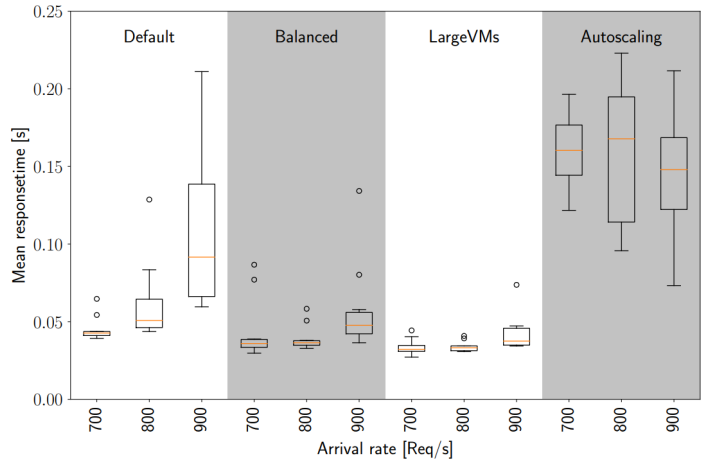


Figure 2: Mean response time for four scenarios and three load-levels each (all distributions consist of ten elements, one for each repetition of the scenario).

Finding 2: The total CPU busy time may not be statistically significantly different between scenarios

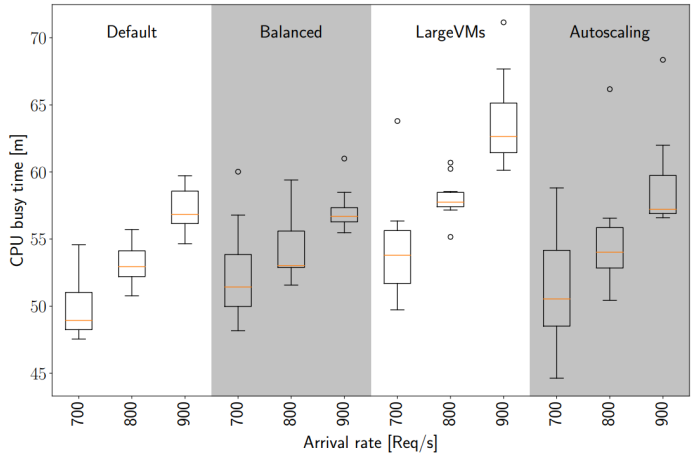


Figure 4: CPU busy time for four scenarios and three load-levels each (N=10).

Research Question 3 – Selected Findings

How well can performance regressions in microservices be detected?

Finding 1: Using only a single experiment run results in flaky performance tests

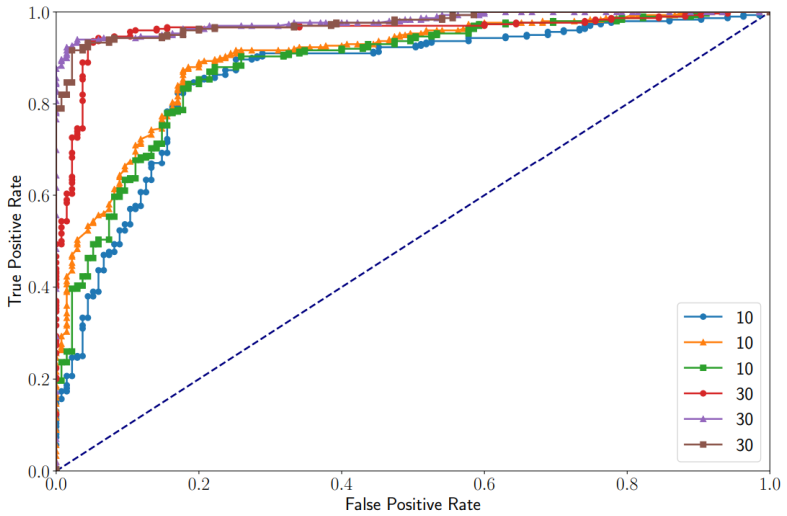


Figure 6: ROC curve showing the detection accuracy for the 10% and 30% regression.

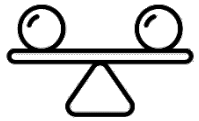
Finding 2: Using ten experiment runs results in stable performance tests

Table 4: Comparing the distributions of the response time and total CPU time between different scenarios with regression between the LargeVMs and 10% Regression scenario (all distributions consist of ten elements, one for each repetition of the scenario—hereafter identified as N=10).

Load [Req/s]	Response time			CPU Utilization		
	p-value	Eff. size		p-value	Eff. size	
700	0.00	1.00	(L)	0.00	1.00	(L)
800	0.00	1.00	(L)	0.00	1.00	(L)
900	0.00	1.00	(L)	0.00	1.00	(L)

Microservices - A Performance Testers Nightmare?

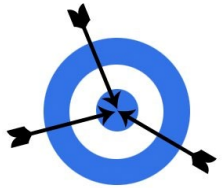
Nightmare 1



Stability of the environment

- Autoscaling/container orchestration is not deterministic
- Execution environment can not be expected to be stable

Nightmare 2



Reproducibility of the experiments

- The repeated experiments may not result in the same performance measurements
- Multiple measurements required for regression testing

Nightmare 3

Detecting small changes

- Variation between measurements can be quite large
- Detecting small changes is challenging

Research Directions

Research Direction 1

Variation reduction in executing performance tests

Research Direction 2

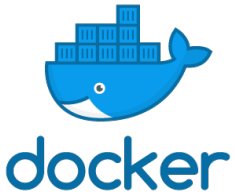
Studying the stability of (new) performance metrics

Research Direction 3

Creating a benchmark environment for microservice-oriented performance engineering research

Replication Package

Performance measurements



Wrapped in docker container for platform independent execution

Requires only Google Cloud access keys as input

Fully automated performance measurements

Available online at:
https://doi.org/10.5281/zenodo.3588515

Data set and analysis

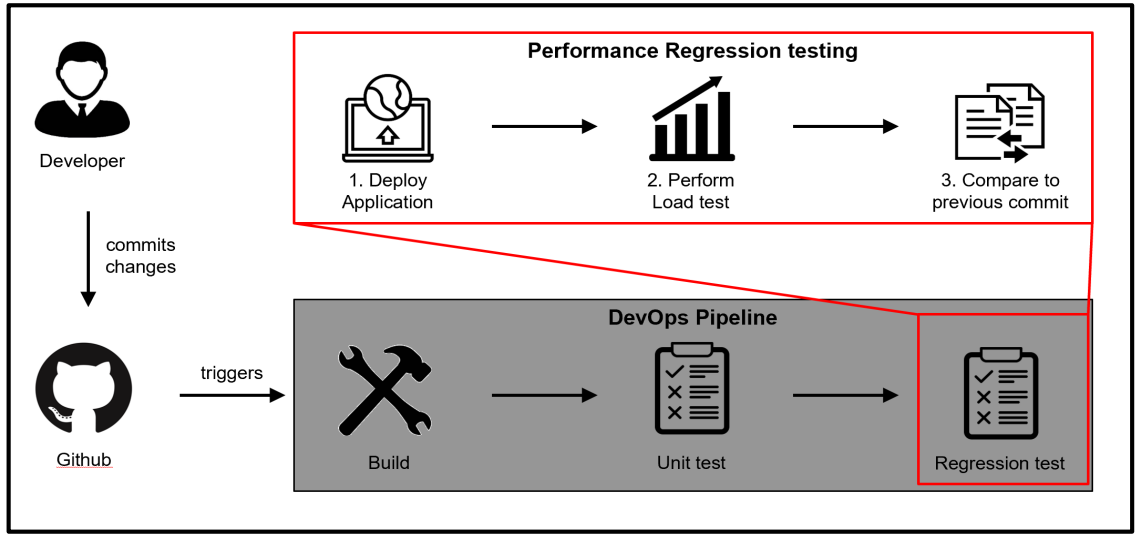
Measurement data of over 75 hours of experiments

Scripts to reproduce any analysis, table or figure from the manuscript

1-click reproduction of the results as a CodeOcean Capsule

Available online at:
https://doi.org/10.24433/CO.4876239.v1

Summary



Microservices - A Performance Testers Dream?

Benefit 1: Containerization

- Containers package environment
- Simplifies setup of test environment

Benefit 2: Granularity

- Individually testable services
- Dependencies via HTTP calls
- Dependencies easily mocked

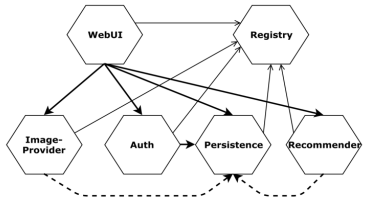
Benefit 3: Easy access to metrics

- Orchestration frameworks simplify metric collection
- Application-level metrics common

Benefit 4: Integration with DevOps

- Size reduces performance test duration
- Performance testing within pipeline

TeaStore Benchmarking Application



Scenarios

Scenario	#Nodes	Cores/Node	Memory/Node
Default	20	1	6.5 GB
Balanced	20	1	6.5 GB
LargeVMs	5	4	26 GB
Autoscaling	5	4	26 GB
Regression (baseline)	5	4	26 GB
Regression	5	4	26 GB

Table 1: Cluster size in the different scenarios.

Deployment Platform

docker kubernetes Terraform Google Cloud Platform

How well can performance regressions in microservices be detected?

Finding 1: Using only a single experiment run results in flaky performance tests

Finding 2: Using ten experiment runs results in stable performance tests

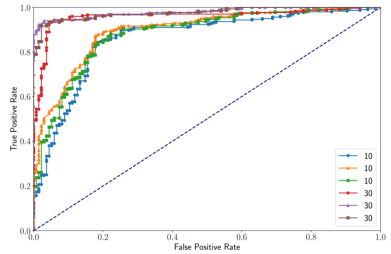


Figure 6: ROC curve showing the detection accuracy for the 10% and 30% regression.

Table 4: Comparing the distributions of the response time and total CPU time between different scenarios with regression between the LargeVMs and 10% Regression scenario (all distributions consist of ten elements, one for each repetition of the scenario—hereafter identified as N=10).

Load [Req/s]	Response time p-value	Response time Eff. size		CPU Utilization p-value	CPU Utilization Eff. size	
700	0.00	1.00	(L)	0.00	1.00	(L)
800	0.00	1.00	(L)	0.00	1.00	(L)
900	0.00	1.00	(L)	0.00	1.00	(L)

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

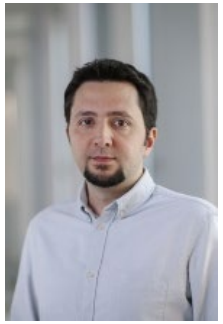
Descartes research

Microservices: A Performance Tester's Dream or Nightmare?

Simon Eismann
University of Würzburg
@simon_eismann

Cor-Paul Bezemer
University of Alberta
@corpaul

Weiyi Shang
Concordia University
@swy351



Dušan Okanović
University of Stuttgart
@okanovic_d

André van Hoorn
University of Stuttgart
@andrevanhoorn

spec

spec RG DevOps Performance

https://research.spec.org/working-groups/rg-devops-performance.html

Artifacts Available acm

Artifacts Evaluated acm Reusable