Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

# Exploring the Feasibility of Load Testing Serverless Applications

**Presentation**

**Simon Eismann**, Diego Costa, Lizhi Liao, Cor-Paul Bezemer,
Weiyi Shang, André van Hoorn, Samuel Kounev

08.11.2020

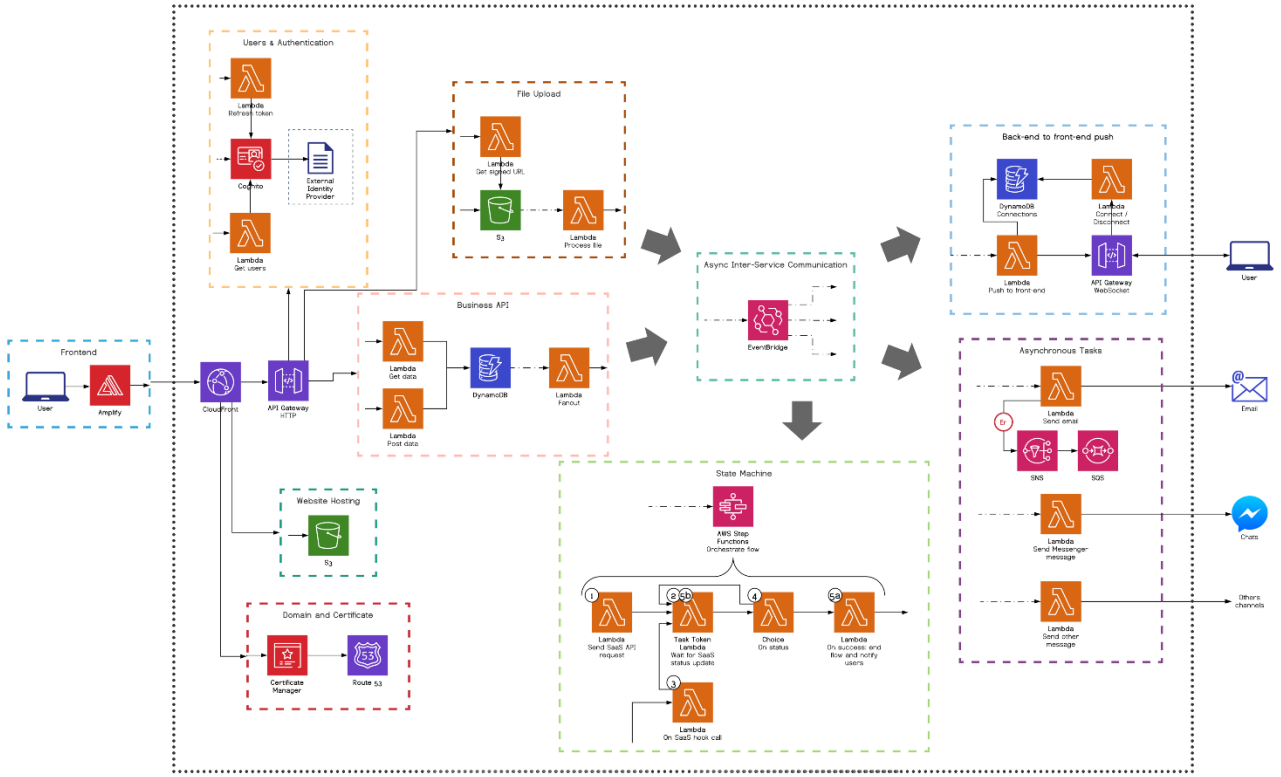
*http://se.informatik.uni-wuerzburg.de/*

spec Research

spec RG DevOps Performance

# What are Serverless Applications?

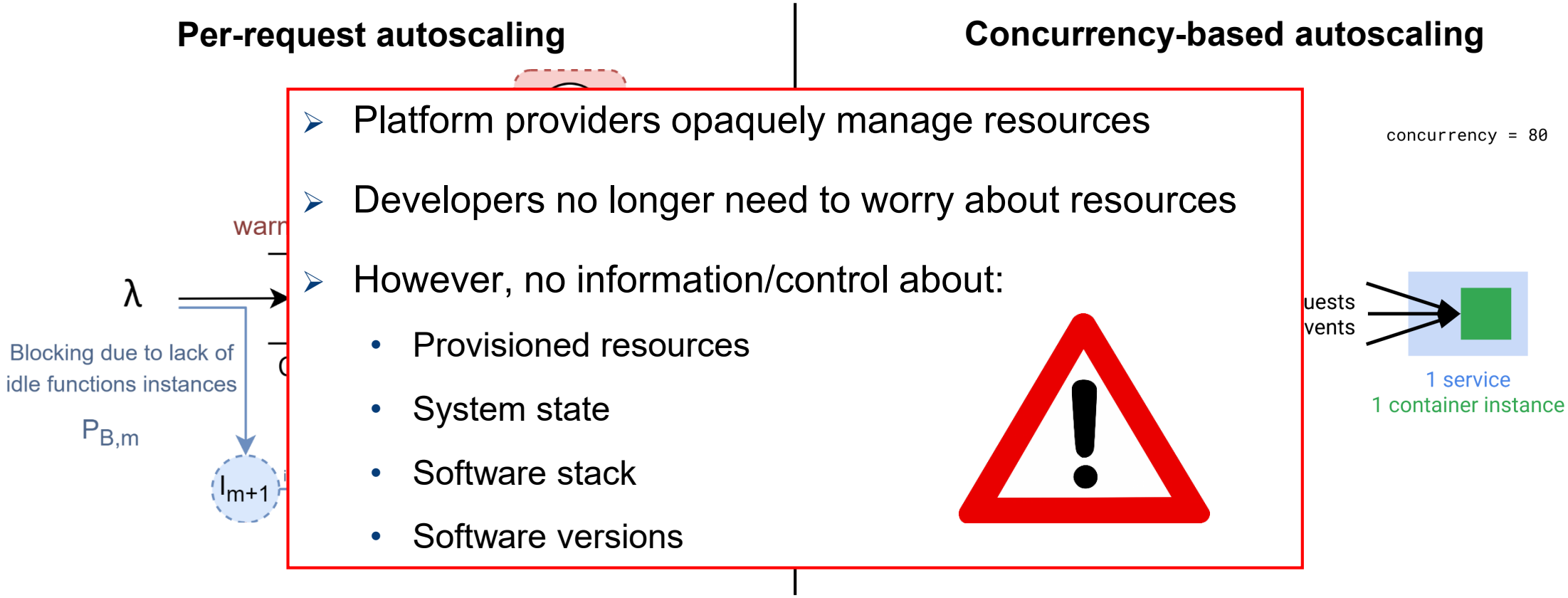
- Serverless = FaaS + BaaS
- Function-as-a-Service (FaaS)
  - Fully managed compute
  - Ephemeral, stateless, shortrunning
- Backend-as-a-Service (BaaS)
  - Fully managed services
  - E.g., database, messaging, auth, …
- Serverless characteristics [1, 2]:
  - Event-driven
  - Granular billing
  - No operational logic




https://medium.com/serverless-transformation/what-a-typical-100-serverless-architecture-looks-like-in-aws-40f252cd0ecb


[1] **Predicting the Costs of Serverless Workflows** Eismann, Simon; Grohmann, Johannes; van Eyk, Erwin; Herbst, Nikolas; Kounev, Samuel; in *Proceedings of the 2020 ACM/SPEC International Conference on Performance Engineering*
[2] **A SPEC RG Cloud Group's Vision on the Performance Challenges of FaaS Cloud Architectures** van Eyk, Erwin; Iosup, Alexandru; Abad, Cristina L.; Grohmann, Johannes; Eismann, Simon; in *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*

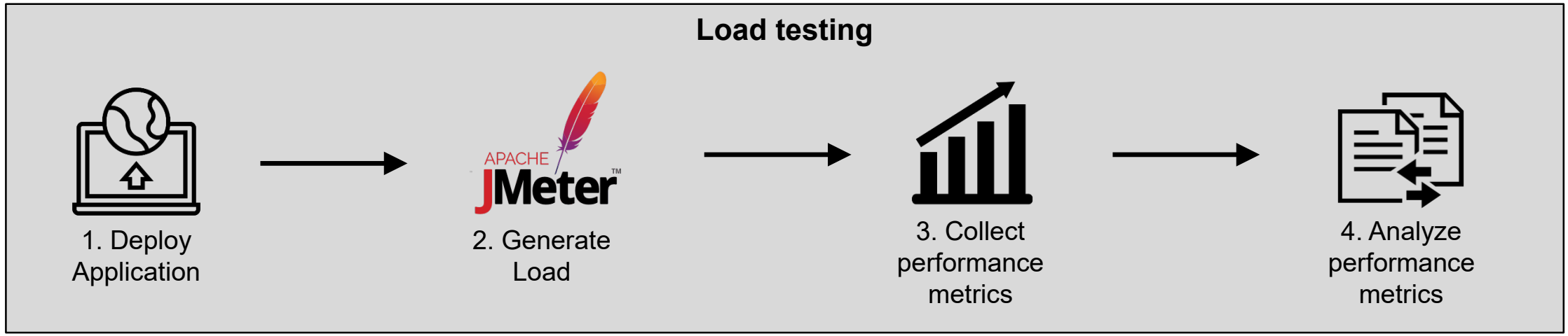
# Resource Management in Serverless Applications




- Platform providers opaquely manage resources
- Developers no longer need to worry about resources
- However, no information/control about:
  - Provisioned resources
  - System state
  - Software stack
  - Software versions

https://itnext.io/autoscaling-patterns-in-serverless-computing-you-should-know-about-9d8c5d00d324
https://cloud.google.com/run/docs/about-concurrency

# Can we load test serverless applications?



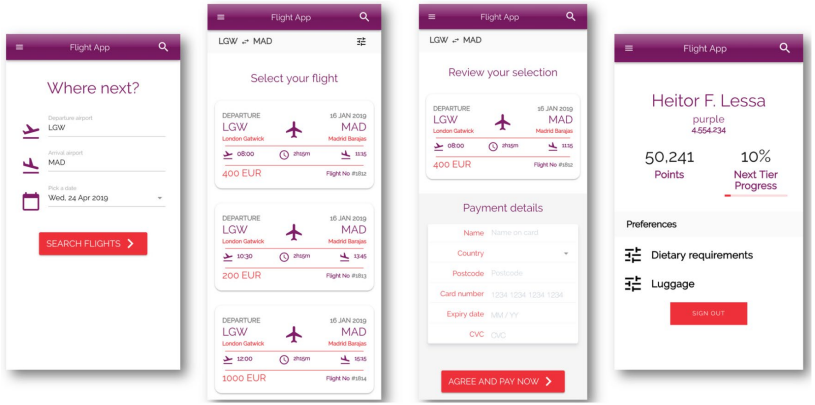

- Common load testing goals:
  - SLA conformance testing ✓
  - Capacity testing ?
  - Regression testing ✓
- Load testing requirements [1]:
  - Representative operational profile ✓
  - Stable testing environment ?
  - Reproducible results ?

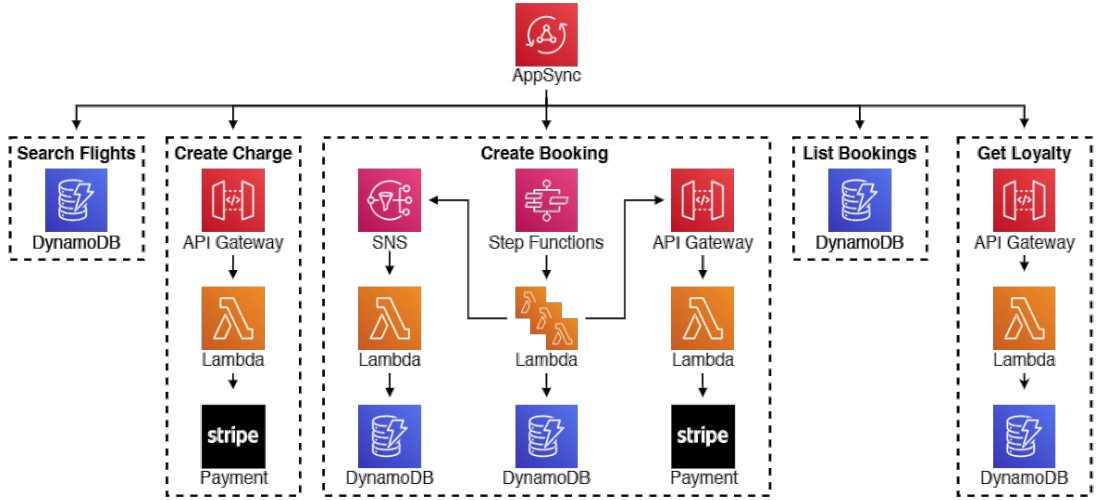
[1] **Microservices: A Performance Tester's Dream or Nightmare?** Eismann, Simon; Bezemer, Cor-Paul; Shang, Weiyi; Okanovic, Dusan; van Hoorn, Andre; in *Proceedings of the 2020 ACM/SPEC International Conference on Performance Engineering*

# Case Study - Serverless Airline Booking Application

## A representative serverless application

- Presented at AWS re:Invent as an example implementation of a production-grade serverless application [2]
- Runs on AWS → the most popular serverless platform [1]
- Implemented in Python/NodeJS → the most popular languages for serverless applications [1]





[1] **Serverless Applications: Why, When, and How?** Eismann, Simon; Joel, Scheuner; van Eyk, Erwin; Schwinger, Maximilian; Grohmann, Johannes; Herbst, Nikolas; Abad, Cristina; Iosup, Alexandru; in *IEEE Software*, 2020
[2] https://www.youtube.com/watch?v=DcrtvgaVdCU

# Case Study - Datasets

## Load testing dataset

- 15 minutes per measurement
- 5/25/50/100/250/500 req/s
- 256/512/1024 MB function sizes
- 10 measurement repetitions

## Longitudinal dataset


- 100 req/s + 512 MB function size
- Daily measurements starting 20 July
- Three repetitions starting 22 Aug
- Still ongoing

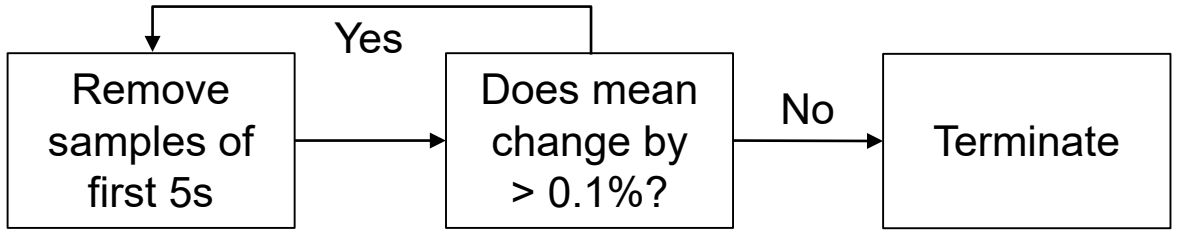
# Research Questions

**RQ1** Is the environment stable within a load test?

**RQ2** Are the load testing results reproducible?

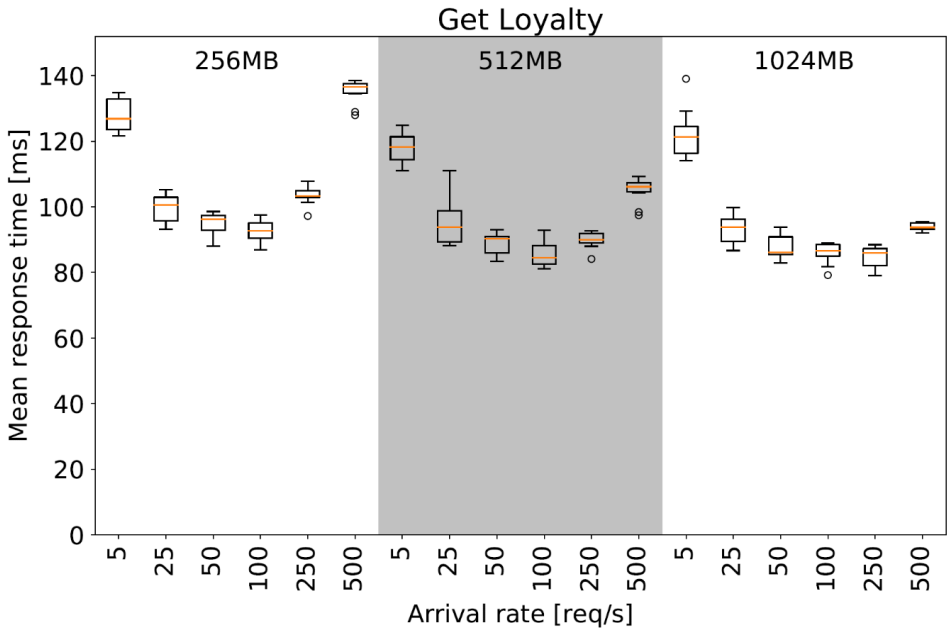
**RQ3** Do platform-side changes impact load testing results?

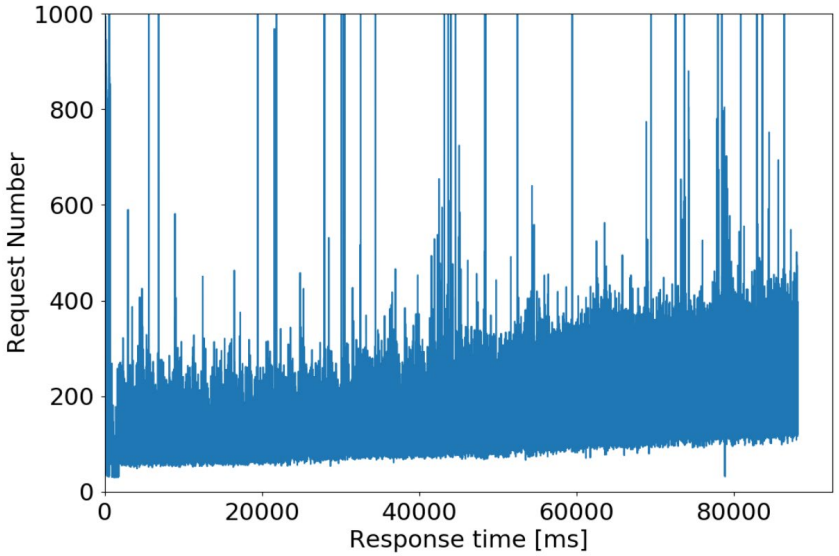
# RQ 1: Is the environment stable within a load test?



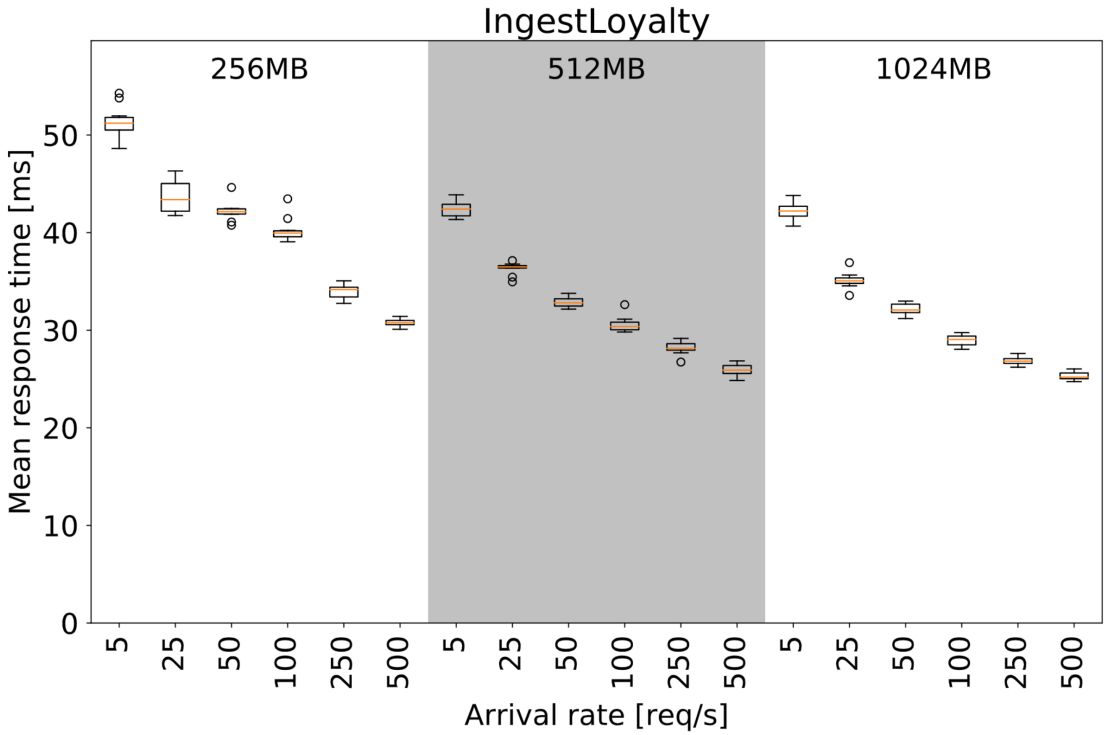

| | 256MB | | | | | | 512MB | | | | | | 1024MB | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 Req/s | 25 Req/s | 50 Req/s | 100 Req/s | 250 Req/s | 500 Req/s | 5 Req/s | 25 Req/s | 50 Req/s | 100 Req/s | 250 Req/s | 500 Req/s | 5 Req/s | 25 Req/s | 50 Req/s | 100 Req/s | 250 Req/s | 500 Req/s |
| CollectPayment | 14.5 | 14.5 | 15.0 | 15.0 | 15.0 | 15.5 | 10.0 | 10.5 | 10.0 | 10.5 | 12.0 | 15.0 | 10.0 | 10.0 | 10.5 | 10.0 | 9.5 | 14.0 |
| CaptureStripeCharge | 14.5 | 13.5 | 15.0 | 15.0 | 15.0 | 15.5 | 10.0 | 11.0 | 10.0 | 10.0 | 12.0 | 15.0 | 6.0 | 6.5 | 8.0 | 7.0 | 8.5 | 10.0 |
| ConfirmBooking | 13.0 | 12.0 | 15.5 | 20.5 | 15.0 | 16.5 | 10.5 | 10.0 | 10.0 | 11.5 | 15.0 | 16.0 | 10.5 | 10.0 | 10.0 | 10.0 | 9.0 | 15.0 |
| CreateStripeCharge | 5.0 | 5.5 | 5.5 | 5.5 | 15.0 | 15.0 | 5.5 | 5.0 | 5.5 | 5.0 | 15.0 | 14.5 | 5.0 | 5.0 | 5.0 | 5.0 | 4.5 | 8.5 |
| Get Loyalty | 39.0 | 41.0 | 31.0 | 16.5 | 10.0 | 660.5 | 52.5 | 40.0 | 29.0 | 16.5 | 10.0 | 17.5 | 37.0 | 36.5 | 27.0 | 17.5 | 9.5 | 15.5 |
| IngestLoyalty | 15.5 | 20.0 | 18.5 | 22.0 | 74.5 | 87.5 | 13.0 | 15.0 | 21.0 | 35.0 | 81.5 | 91.5 | 11.0 | 12.5 | 15.0 | 34.5 | 81.5 | 98.5 |
| List Bookings | 69.0 | 52.0 | 41.0 | 22.0 | 12.0 | 8.5 | 68.0 | 49.0 | 36.5 | 24.5 | 10.0 | 7.0 | 59.5 | 52.5 | 35.0 | 25.0 | 10.5 | 6.5 |
| NotifyBooking | 16.0 | 22.0 | 20.5 | 19.0 | 20.0 | 19.5 | 11.5 | 10.5 | 11.5 | 13.0 | 15.0 | 16.5 | 11.0 | 10.0 | 11.5 | 10.0 | 13.0 | 15.5 |
| Process Booking | 25.5 | 13.0 | 10.5 | 7.5 | 5.5 | 5.0 | 20.5 | 17.5 | 9.5 | 6.0 | 5.5 | 5.0 | 18.0 | 17.5 | 9.5 | 6.0 | 5.0 | 5.0 |
| ReserveBooking | 5.5 | 5.5 | 6.0 | 8.0 | 10.0 | 13.5 | 6.0 | 6.0 | 7.0 | 10.0 | 10.0 | 10.0 | 5.5 | 5.0 | 5.0 | 5.0 | 9.0 | 10.0 |
| CollectPayment | 14.5 | 14.5 | 15.0 | 15.0 | 15.0 | 15.5 | 10.0 | 10.5 | 10.0 | 10.5 | 12.0 | 15.0 | 10.0 | 10.0 | 10.5 | 10.0 | 9.5 | 14.0 |
| Search Flights | 64.0 | 54.5 | 37.5 | 22.5 | 11.0 | 7.0 | 82.0 | 51.0 | 33.0 | 22.5 | 10.0 | 7.0 | 64.5 | 52.5 | 34.5 | 24.0 | 10.5 | 7.0 |

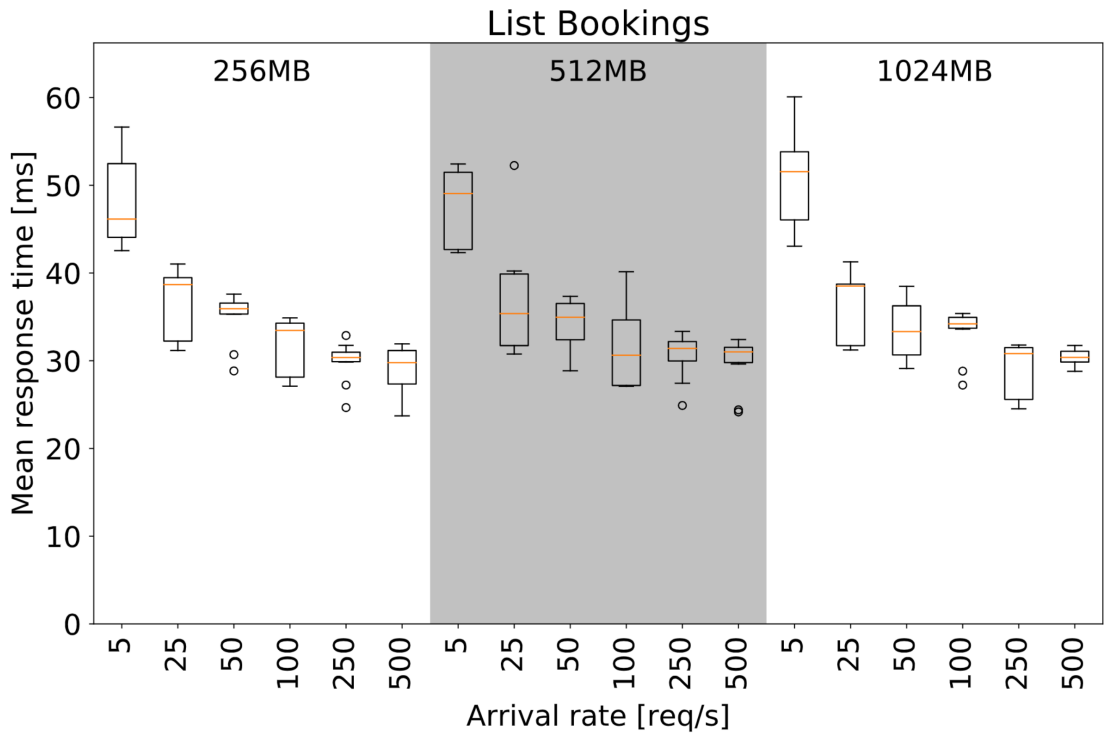
# Warmup - Outlier at GetLoyalty/256MB/500req/s

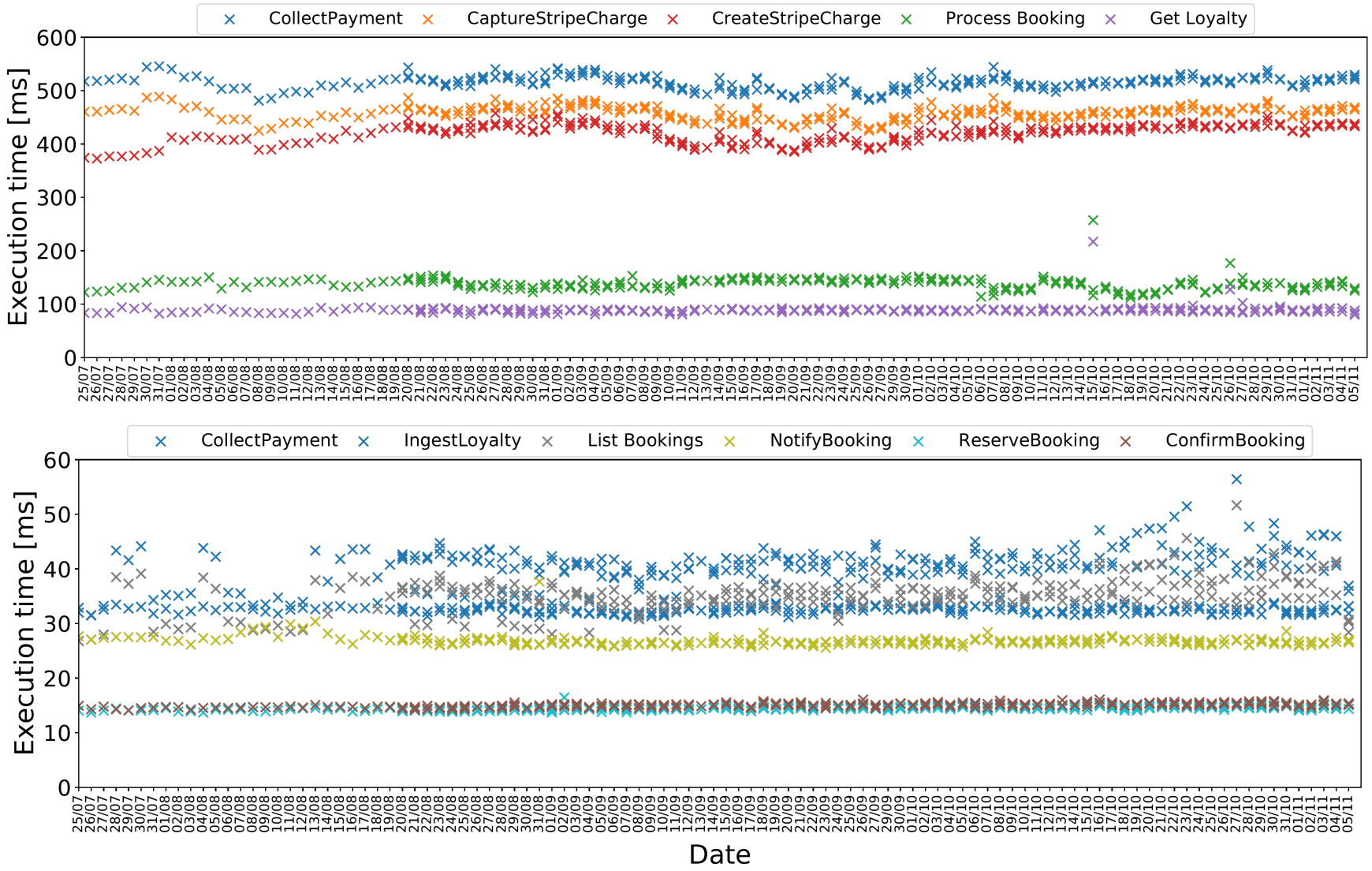



# RQ 2: Are the load testing results reproducible?



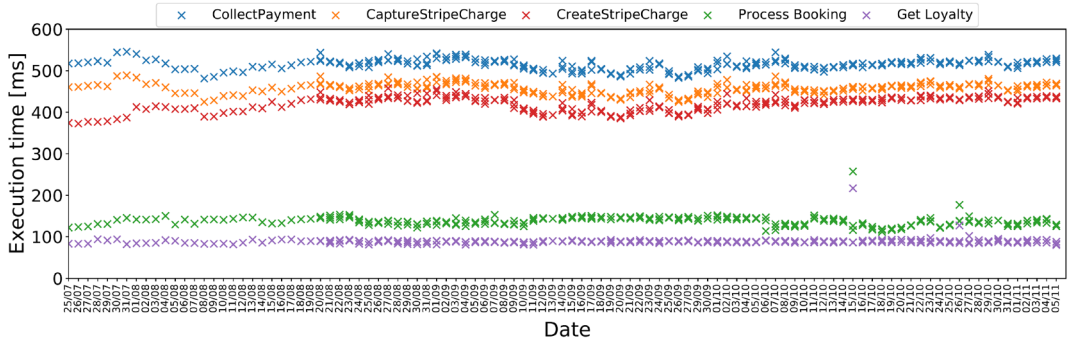



# RQ3: Do platform-side changes impact load testing results?



# Interested in performance regression testing?

- This work is conducted in the context of the SPEC RG DevOps
- Ongoing Activities:
  - Performance regression testing of modern applications
  - Model extraction in continuous software engineering
  - Performance of continuous delivery infrastructures
- New activity: Automated detection of performance regressions




- Interested? Join our regular calls at: https://research.spec.org/en/working-groups/rg-devops-performance.html