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/
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

[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
Resource Management in Serverless Applications

Per-request autoscaling

- 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

Concurrency-based autoscaling

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?

Load testing

1. Deploy Application
2. Generate Load
3. Collect performance metrics
4. Analyze performance metrics

- **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]
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
Exploring the Feasibility of Load Testing Serverless Applications

Simon Eismann

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?

Remove samples of first 5s

Yes

Does mean change by > 0.1%?

No

Terminate

Exploring the Feasibility of Load Testing Serverless Applications

Simon Eismann
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