ORCAS: Efficient Resilience Benchmarking of Microservice Architectures

André van Hoorn
Aldeida Aleti
Thomas F. Düllmann
Teerat Pitakrat

9th Symposium on Software Performance (SSP 2018)
November 08, 2018. Hildesheim
Previously Presented at ISSRE 2018

The 29th IEEE International Symposium on Software Reliability Engineering (ISSRE 2018), October 15-18, 2018
Resilience Antipattern

“Recurring solution to common problem with negative consequences for the system”

Resilience Pattern Example: Circuit Breaker
Resilience Benchmarking – aka Chaos Engineering

• How to accept failures? – Learning by doing:
  Intentionally inject failures into the production system

  “Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system’s capability to withstand turbulent conditions in production.” — Principles of Chaos

• Who is doing this?

  Game Day exercises

  Simian Army for AWS

van Hoorn et al.: Efficient Resilience Benchmarking of Microservice Architectures
“Current resilience benchmarking practice is inefficient.”

André et al.

Goal: **Make it more efficient!**
Idea of the \textsc{orcas} Project

- Leverage relationship between resilience patterns, antipatterns, and fault injections
- Consider software architectural knowledge to generate experiments
- Combine model-based (simulations) and measurement-based ("real") resilience experiments
Envisioned Framework

- Static and dynamic analysis + manual enrichment

System → extraction → Architectural Information

- Services, deployment, (remote) interactions
- Patterns and anti-patterns
- Criticality of services
- Steady-state metrics
Envisioned Framework

Static and dynamic analysis + manual enrichment

System → extraction → Architectural Information → input → Orcas Decision Engine

- Services, deployment, (remote) interactions
- Patterns and anti-patterns
- Criticality of services
- Steady-state metrics

Knowledge and algorithms – „the magic“
Envisioned Framework

System → extraction → Architectural Information → input → Orcas Decision Engine

Experiment real
- Workload
- Faultload

execution

results

generation
Envisioned Framework

System ➔ extraction ➔ Architectural Information ➔ input ➔ Orcas Decision Engine

System Simulation ➔ input ➔ Experiment real ➔ Workload ➔ Faultload

Experiment sim ➔ Workload ➔ Faultload ➔ generation ➔ results

text: van Hoorn et al.: Efficient Resilience Benchmarking of Microservice Architectures
Resilience Benchmarking – aka Chaos Engineering

- How to accept failures? – Learning by doing:
  Intentionally inject failures into the production system

  "Chaos Engineering is the discipline of experimenting on a
distributed system in order to build confidence in the
system’s capability to withstand turbulent conditions in
production."
  — Principles of Chaos

- Who is doing this?
  - Game Day exercises
  - Simian Army for AWS
  - Error Monkey for Node.js

Current and Next Steps

- PoC implementation
- Evaluation of injection frameworks
- Simulator extensions
- Developing "the magic"
- Experimental evaluation
- Industry case study (?)
5th International Workshop on Quality-Aware DevOps (QUDOS 2019) Joint with the 4th Workshop on Continuous Software Engineering

QUDOS 2019 will be held on Mar 26th, 2019 in Hamburg, Germany and will be co-located with the 2nd International Conference on Software Architectures (ICSA 2019).