Extracting Software Architecture from Traces for the Simulation of Microservice Architectures

SSP 22 – Extended Abstract

Tim Thüring, Gabriel Glaser, Abel Gitzing, Marcel Hafner, Sebastian Frank, Alireza Hakamian and André van Hoorn
Extracting Software Architecture from Microservices for Simulation

**MiSim:**
Simulates Microservice architectures with focus on resilience

[Sebastian Frank, Lion Wagner, Alireza Hakamian, Martin Straesser, André van Hoorn: MiSim: A Simulator for Resilience Assessment of Microservice-based Architectures. QRS 2022. Accepted]

**RESIRIO:**
Supports engineers during the requirements specification

Extracting Software Architecture from Microservices for Simulation

How to obtain such an architecture description automatically?

Software Architecture Extraction (SAE)

Dependency

Resilience Patterns

Microservice-System

Service

Automate the Process

Architecture Description

Simulation and Analysis of the Microservice

MiSim and RESIRIO

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al.

08.11.2022
Extracting Software Architecture from Microservices for Simulation

How to obtain information about the architecture from traces?

Jaeger, Zipkin → Traces → SAE tool → Architecture Description → Simulation and Analysis of the Microservice

MiSim and RESIRIO

Microservice-System
Extracting Software Architecture from Microservices for Simulation

Microservice-System

Zipkin trace

MiSim architecture description

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al.
Extracting Software Architecture from Microservices for Simulation

Microservice-System

Zipkin trace

```
{
    "traceId": "6ecb7352e359d33b",
    "parentId": "6eb7352e359d33b",
    "id": "71067bc1e862ddae",
    "name": "get /b1", [...]
    "localEndpoint": {
        "serviceName": "b",
        "ipv4": "172.20.0.5"
    },
    "remoteEndpoint": {"ipv4": "172.20.0.6"...}
}
```

MiSim architecture description

```
{
    "name": "b",
    "instances": 1, [...]
    "patterns": [
        {
            "type": "retry",
            "strategy": {
                "type": "exponential",
                "config": {
                    "baseBackoff": 0.05051483889426919,
                    "base": 2.4928600434324386})
        ],
        "operations": [?
            {
                "name": "get /b1",
                "demand": 188,
                "dependencies": [
                    {
                        "service": "d",
                        "operation": "get /d1"...
                    }
                ]
            }
        ]
    }
}
```
Extracting Software Architecture from Microservices for Simulation

Microservice-System

Zipkin trace

Estimation and Heuristics

MiSim architecture description

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al. 08.11.2022
Development of an SAE tool that automates the SAE process and…

- is compatible with MiSim and RESIRIO
- offers additional features: round-robin load balancer detection, retry detection
- provides Resource-Demand-Estimation (RDE)
- is compatible with more input formats like DynaTrace or Kieker
Extraction Process

- Jaeger Traces
- Zipkin Traces
- DynaTrace
- Kieker
- ...

Open. xtrace

Unifies several trace formats

New version of the SAE tool

Service A
- Operation 1
- Operation 2

Service B
- Operation 1

Generic Trace Representation

Retry Detection
Round-Robin Load Balancer Detection
RDE

Output Generation

RESIRIO architecture description

MiSim architecture description

- LibReDE


Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al.

08.11.2022
Overview of the SAE Tool – Retry Detection

Detect a Retry – Situation in Trace data

Analyze wait-times

Support for linear and exponential backoff-functions

Estimated Retry Configuration

Detect a Retry

Support for linear and exponential backoff-functions

Estimated Retry Configuration

Analyze wait-times

Support for linear and exponential backoff-functions

Estimated Retry Configuration

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al. 08.11.2022 5
Overview of the SAE Tool – Round-Robin Load Balancer Strategy Detection

Distribution of Requests

Load Balancer

Incoming Requests

Instances of a service

Implementation of a Heuristic to Detect a Round-Robin Strategy

Round-Robin Load Balancer Strategy

Count the number of errors in the pattern

Small errors in order can occur

Round-Robin Strategy if error-rate < 10%

Real World

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al.

08.11.2022
Overview of Evaluation

**Retry detection**
- Can the SAE tool estimate configuration parameters?

**Round-robin load balancer detection**
- Can the SAE tool detect a round-robin load balancing strategy?

**Resource demand estimation**
- Are the calculated resource demands representing the actual situation in a system correctly?

**Real-World Traces**
- Can the SAE tool handle traces from a real-world system?

Sample Microservice-System

- One operation with heavy workload
- Round-Robin load balancing with HAProxy
- Zipkin-Traces

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al. 08.11.2022
Results – Retry Detection

Backoff function: Wait time for x-th retry

Resilience4j Configuration

$t(x) = 0.05 \cdot 2.5^x$

Estimation of configuration parameters

Extracted Configuration

$t(x) \approx 0.05 \cdot 2.49^x$
Results – Round-Robin Load Balancer Strategy Detection

5 Instances of one Service:
Instances 1 to 5 of the service

- Detected Error in Pattern
- Detected Upscaling
- Detected Downscaling

- 7 Errors, 143 Entries
  \[
  \frac{7}{143} \approx 0.05 < 0.1
  \]

Threshold

Position in the sequence of instances

Extracting software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al. 08.11.2022
Overview

SAE tool

Generic Trace Representation

- Jaeger Traces
- Zipkin Traces
- OPEN.xtrace
- DynaTrace
- Kieker
- ...

Output Generation

RESIRIO architecture description

MiSim architecture description

Retry Detection
Round-Robin Load Balancer Detection
RDE

Model Refinement

Available on GitHub

DynaTrace
Kieker
...

LibReDE

Retrieving software architecture from traces for the simulation of microservice architectures - Abel G., Gabriel G., Marcel H., Tim T. et al.

All Icons by flaticon.com
Sources of Pictures and Icons

- Resilience4j Logo (Slides 2,7): https://github.com/resilience4j
- Zipkin Logo (Slide 7): https://github.com/openzipkin

- All Icons by Flaticon.com. Individual authors:
  - Simulation (Slide 2): xnimrodx
  - Architecture File (Slide 2, modified): Freepik
  - Goal 1 (Slides 3,4,10): Freepik
  - Goal 2 (Slides 3,4,5,6,8,9,10): fjstudio
  - Goal 3 (Slides 3,4,10): Freepik (Stopwatch) and Flat Icons (CPU)
  - Goal 4 (Slides 3,4,10): noomtah
  - Checkmark (Slides 5,6,9,10): Alfredo Hernandez
  - Cross (Slide 5): Alfredo Hernandez
  - Wait Symbol (Slide 5): Freepik