

Porkins: Eine Instrumentalisierungsbibliothek zur Generierung von Tracingevents von streambasierten Anwendungen

BACHLORTHESES/SEMINARARBEIT
Studiengang Informatik

vorgelegt von
Simon Stockhause

April 2020

Referent der Arbeit:	Prof. Dr. Harald Ritz
Koreferent der Arbeit:	Prof. Dr. Harald Ritz
Betreuer:	M.Sc. Pascal Bormann

Zusammenfassung

Die heutigen Bedürfnisse der Anwender ein stets erreichbaren Service zur Verfügung zu haben, stellt hohe Erwartung an Unternehmen und damit hohe Erwartungen an die Infrastruktur ihrer Mircoserviceanwendungen. Die enorme Skalierbarkeit einzelner Komponenten und die ausgezeichnete Ressourcennutzung der Hardware löst viele Probleme der Vergangenheit. Allerdings schafft diese Umstellung neue Herausforderungen die es zu bewältigen gilt.

- Überwachung
- Logs, Metrics, Traces
- Performance

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	1
1.3	Forschungsstand	1
1.4	Thesisübersicht	1
2	Themenüberblick	2
2.1	Verteilte Systeme	2
2.1.1	Überwachung von verteilten Systemen	2
2.1.2	Synchronisation	2
2.1.3	Ordnung von events	2
2.2	Bibliotheksentwicklung	2
3	Problembeschreibung	3
3.1	Eventgenerierung	3
3.1.1	Eventkorrelation	3
3.1.2	Synchronisation von Eventgeneratoren	3
3.2	Eventübermittlung	3
4	Design	4
4.1	Datenmodell	4
4.1.1	Eventmodell	4
4.1.2	Eventgraph	4
4.2	Verarbeitungsmodell	4
4.2.1	Agenten	4
4.2.2	Collectoren	4
5	Implementierung	5
5.1	Bibliothek: Porkins	5
5.2	Eventagent	5
5.3	Eventcollector	5
6	Evaluierung	6
6.1	Genauigkeit der Eventgenerierung	6

6.2	Darstellung der Events	6
6.3	Vergleich mit Jaeger	6
6.3.1	Datenmodelle	6
6.3.2	Bereitstellung	6
6.3.3	Ergebnisse	6
	Fazit	7
	Literatur	8

Abbildungsverzeichnis

1 | Einleitung

1.1 Motivation

1.2 Problemstellung

1.3 Forschungsstand

1.4 Thesisübersicht

2 | Themenüberblick

2.1 Verteilte Systeme

2.1.1 Überwachung von verteilten Systemen

2.1.2 Synchronisation

2.1.3 Ordnung von events

2.2 Bibliotheksentwicklung

3 | Problembeschreibung

3.1 Eventgenerierung

3.1.1 Eventkorrelation

3.1.2 Synchronisation von Eventgeneratoren

3.2 Eventübermittlung

4 | Design

4.1 Datenmodell

4.1.1 Eventmodell

4.1.2 Eventgraph

4.2 Verarbeitungsmodell

4.2.1 Agenten

4.2.2 Collectoren

5 | Implementierung

5.1 Bilbiothek: Porkins

5.2 Eventagent

5.3 Eventcollector

6 | Evaluierung

6.1 Genauigkeit der Eventgenerierung

6.2 Darstellung der Events

6.3 Vergleich mit Jaeger

6.3.1 Datenmodelle

6.3.2 Bereitstellung

6.3.3 Ergebnisse

Fazit

Literatur

- [Dra+17] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazza-
ra, Fabrizio Montesi, Ruslan Mustafin und Larisa Safina. *Present and Ulterior
Software Engineering*. 2017.