

Generierung und Ordnung von Events in verteilten Systemen mit asynchroner Kommunikation

BACHLORTHESES
Studiengang Informatik

vorgelegt von
Simon Stockhause

Mai 2020

Referent der Arbeit: Prof. Dr. Harald Ritz
Korreferent der Arbeit: 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	Anforderungsanalyse	4
4.1.1	Anforderungen	4
4.1.1.1	Funktionale Anforderungen	4
4.1.1.2	Nicht-Funktionale Anforderungen	4
4.2	Datenmodell	4
4.2.1	Eventmodell	4
4.2.2	Eventgraph	4
4.3	Verarbeitungsmodell	4
4.3.1	Agenten	4
4.3.2	Collectoren	4
5	Implementierung	5
5.1	Bibliothek: Traktor	5
5.2	Traktor Agent	5

5.3	Traktor Registry	5
6	Evaluierung	6
6.1	Genauigkeit der Eventgenerierung	6
6.1.1	Uhren und Zeit	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
7	Fazit	7
7.1	Ausblick	7
7.2	ZitatTest	7
	Literatur	8

Abbildungsverzeichnis

1 | Einleitung

1.1 Motivation

Glossarbeispiel [latex](#)

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 Anforderungsanalyse

4.1.1 Anforderungen

4.1.1.1 Funktionale Anforderungen

4.1.1.2 Nicht-Funktionale Anforderungen

4.2 Datenmodell

4.2.1 Eventmodell

4.2.2 Eventgraph

4.3 Verarbeitungsmodell

4.3.1 Agenten

4.3.2 Collectoren

5 | Implementierung

5.1 Bilbiothek: Traktor

5.2 Traktor Agent

5.3 Traktor Registry

6 | Evaluierung

6.1 Genauigkeit der Eventgenerierung

6.1.1 Uhren und Zeit

6.2 Darstellung der Events

6.3 Vergleich mit Jaeger

6.3.1 Datenmodelle

6.3.2 Bereitstellung

6.3.3 Ergebnisse

7 | Fazit

7.1 Ausblick

7.2 ZitatTest

[MRF15] [Sam+16] [MF18] [Bar+04] [Rey+06] [ACF12] [NCK19] [Bey+16] [Kal+17] [Bar+03]
[Red] [SCR18] [Sig+10] [Ste01] [Fon+07] [Wat17] [Ope19]

Literatur

- [ACF12] Mona Attariyan, Michael Chow und Jason Flinn. „X-Ray: Automating Root-Cause Diagnosis of Performance Anomalies in Production Software“. In: *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*. OSDI'12. USA: USENIX Association, 2012, S. 307–320. ISBN: 9781931971966.
- [Bar+03] P Barham, R Isaacs, R Moertier und D Narayanan. „Magpie: online modeling and performance-aware systems“. In: *Proceedings of USENIX HotOS IX* (2003).
- [Bar+04] Paul Barham, Austin Donnelly, Rebecca Isaacs und Richard Mortier. „Using Magpie for Request Extraction and Workload Modelling“. In: *Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6*. OSDI'04. USA: USENIX Association, 2004, S. 18.
- [Bey+16] B Beyer, C Jones, J Petoff und N R Murphy. *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, Incorporated, 2016. ISBN: 9781491929124. URL: <https://books.google.de/books?id=81UrjwEACAAJ>.
- [Fon+07] R Fonseca, G Porter, R Katz, S Shenker und I Stocia. „X-Trace: A Pervasive Network Tracing Framework“. In: *Proceedings of USENIX NSDI* (2007).
- [Kal+17] J Kaldor, J Mace, M Bejda, E Gao, W Kuropatwa, J O'Neill, K Win Ong, B Schaller, P Shan, B Viscomi, V Venkataraman, K Veeraraghavan und Y Jiun Song. „Canopy: An End-to-End Performance Tracing And Analysis System“. In: *SOPS 2017* (2017).
- [MF18] Jonathan Mace und Rodrigo Fonseca. „Universal Context Propagation for Distributed System Instrumentation“. In: *Proceedings of the Thirteenth EuroSys Conference*. EuroSys '18. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450355841. DOI: [10.1145/3190508.3190526](https://doi.org/10.1145/3190508.3190526). URL: <https://doi.org/10.1145/3190508.3190526>.
- [MRF15] Jonathan Mace, Ryan Roelke und Rodrigo Fonseca. „Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems“. In: *Proceedings of the 25th Symposium on Operating Systems Principles*. SOSP '15. New York, NY, USA: Association for Computing Machinery, 2015, S. 378–393. ISBN: 9781450338349. DOI: [10.1145/2815400.2815415](https://doi.org/10.1145/2815400.2815415). URL: <https://doi.org/10.1145/2815400.2815415>.

-
- [NCK19] S Nedelkoski, J Cardoso und O Kao. „Anomaly Detection and Classification using Distributed Tracing and Deep Learning“. In: *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. 2019, S. 241–250. DOI: [10.1109/CCGRID.2019.00038](https://doi.org/10.1109/CCGRID.2019.00038).
- [Ope19] OpenTracing. *opentracing trace overview figure*. 2019. URL: <https://opentracing.io/docs/overview/>.
- [Red] Inc Red Hat. *Was ist IT-Automatisierung?* URL: <https://www.redhat.com/de/topics/automation/whats-it-automation>.
- [Rey+06] Patrick Reynolds, Charles Killian, Janet L Wiener, Jeffrey C Mogul, Mehul A Shah und Amin Vahdat. „Pip: Detecting the Unexpected in Distributed Systems“. In: *Proceedings of the 3rd Conference on Networked Systems Design & Implementation - Volume 3*. NSDI’06. USA: USENIX Association, 2006, S. 9.
- [Sam+16] Raja R Sambasivan, Ilari Shafer, Jonathan Mace, Benjamin H Sigelman, Rodrigo Fonseca und Gregory R Ganger. „Principled Workflow-Centric Tracing of Distributed Systems“. In: *Proceedings of the Seventh ACM Symposium on Cloud Computing*. SoCC ’16. New York, NY, USA: Association for Computing Machinery, 2016, S. 401–414. ISBN: 9781450345255. DOI: [10.1145/2987550.2987568](https://doi.org/10.1145/2987550.2987568). URL: <https://doi.org/10.1145/2987550.2987568>.
- [SCR18] MARIO SCROCCA. „Towards observability with (RDF) trace stream processing“. Diss. Politecnico Di Milano, 2018. URL: <http://hdl.handle.net/10589/144741>.
- [Sig+10] Benjamin Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan und Chandan Shanbhag. „Dapper, a Large-Scale Distributed Systems Tracing Infrastructure“. In: *Google Technical Report dapper-2010-1* (2010).
- [Ste01] William Stearns. *Ngrep and regular expressions to the rescue*. <http://www.stearns.org/>. 2001. URL: <http://www.stearns.org/doc/ngrep-intro.current.html>.
- [Wat17] Matt Watson. *8 Key Application Performance Metrics & How to Measure Them*. 2017. URL: <https://stackify.com/application-performance-metrics/>.