TERRITORI APERTI

SOBIGDATA RESEARCH INFRASTRUCTURE

Tecniche di Visualizzazione per l'Analisi di Problematiche di Prestazioni nei Sistemi a Microservizi

Luca Traini
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

COMUNE DELL'AQUILA

PARTNER

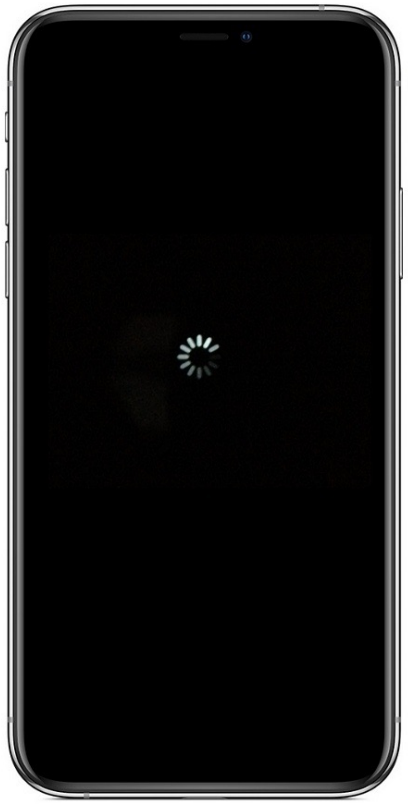
CNR-ISTI

USRA Ufficio Speciale per la Ricostruzione dell'Aquila

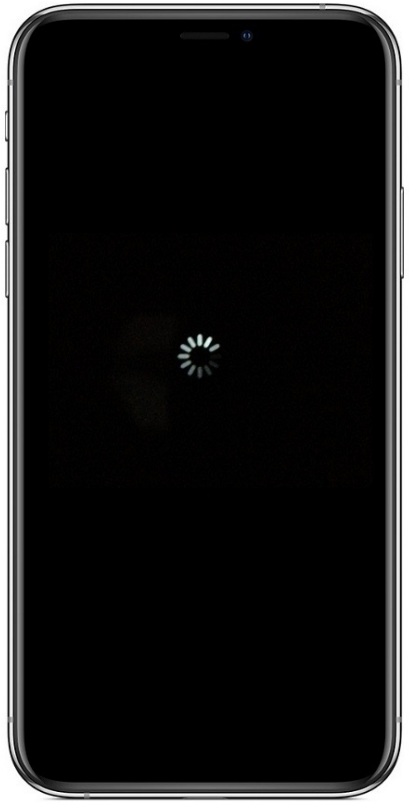
UNIVERSITA' DEGLI STUDI D'ANNUNZIO

CGIL CISL UIL

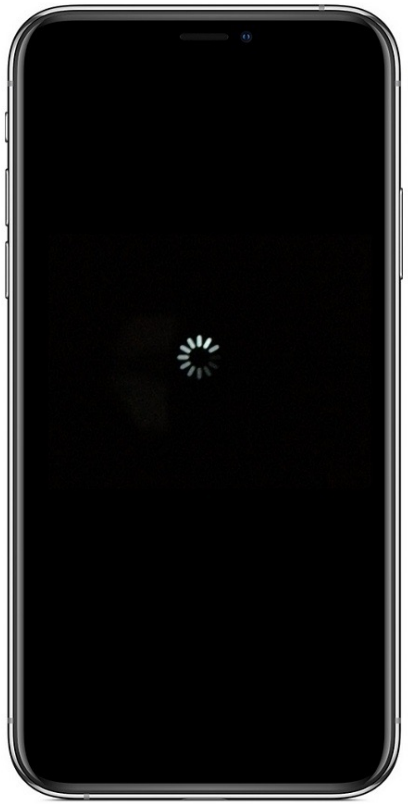
Software Performance Issues



Software Performance Issue and Natural Disasters



Software Performance Issue and Risk Management



WIRED SCIENZA ECONOMIA CULTURA MORE EVENTI NEWSLETTER MAGAZINE

KEVIN CARBONI DOMANDE & RISPOSTE 27.06.2023

Come funziona It-alert, il nuovo sistema nazionale di allerta

Gestito dalla Protezione civile, invia una
interessata da un pericolo. A

The Verge / Tech / Reviews / Science / Entertainment / More

The Verge

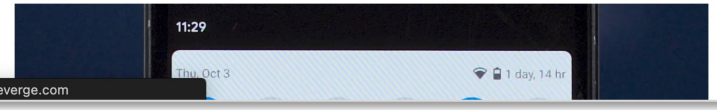
AD efined

SCIENCE / ENVIRONMENT

California's earthquake warning app: how it works and what comes next / Smartphones will warn Californians of an earthquake seconds before they feel them

By Justine Calma, a science reporter covering the environment, climate, and energy with a decade of experience. She is also the host of the Hell or High Water podcast.

Oct 17, 2019, 11:16 PM GMT+2 | 0 Comments / 0 New



v.theverge.com

Software Performance Assurance

- Proactive Approach
  - Identifying Performance Issues Before Software Release
  - Pre-production Software Performance Testing
- Reactive Approach
  - Rapid Identification of Performance Issues in Production
  - Analyzing System Telemetry Data for Root Cause Analysis

Software Performance Assurance

- Proactive Approach
  - Identifying Performance Issues Before Software Release
  - Pre-production Software Performance Testing

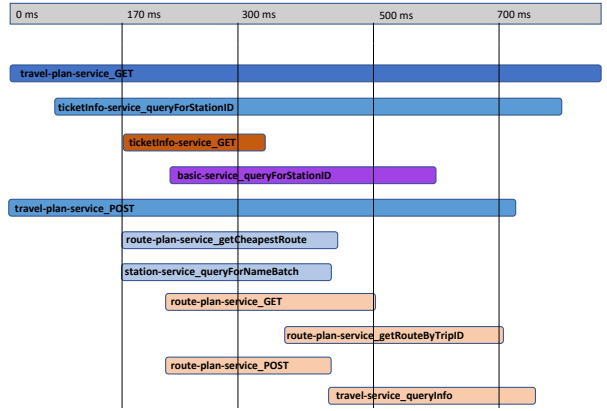
- Reactive Approach
  - Rapid Identification of Performance Issues in Production
  - Analyzing System Telemetry Data for Root Cause Analysis

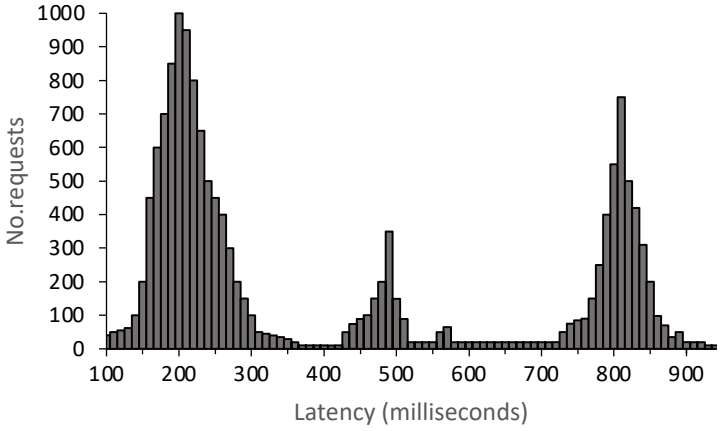
Challenges in Reactive Approaches

- Modern Software Systems Generate a Substantial Volume of Telemetry Data
- Pinpointing the Root Cause of Software Performance Issues Can Be a Complex Task

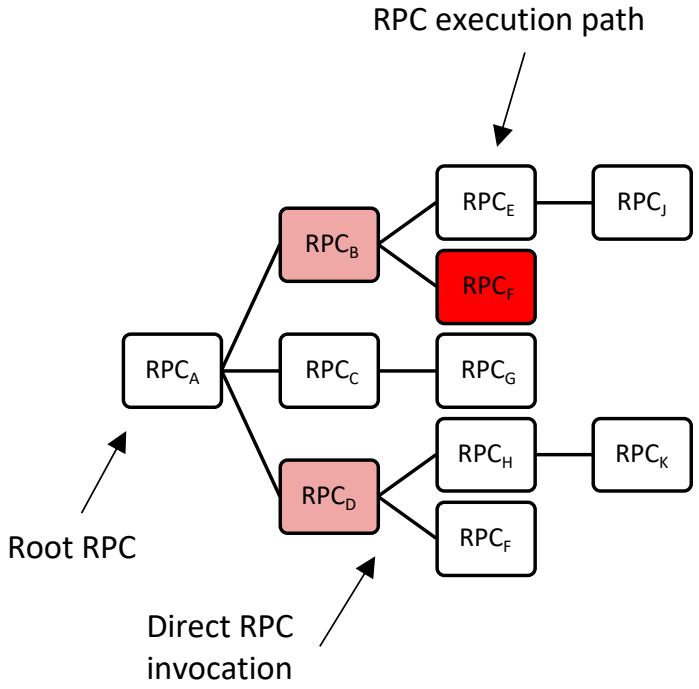
Data Visualization for Software Performance Analysis

- Visualizations Designed for Performance Analysis of Individual Software Executions (Fine-Grained Analysis)
- Visualizations for Performance Analysis of Aggregated Telemetry Data (Coarse-Grained Analysis)
- Bridging the Gap Between Fine-Grained and Coarse-Grained Analysis

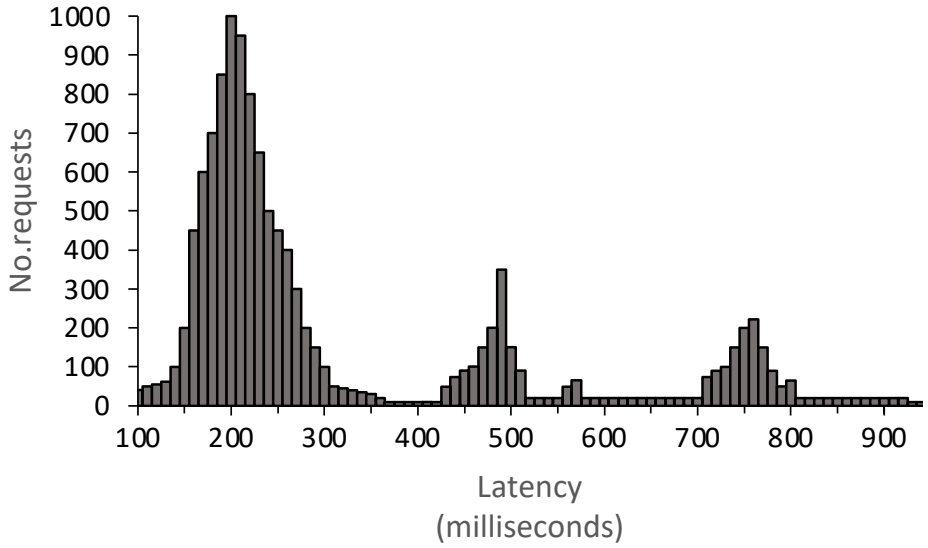




VAMP: Visual Analytics for Microservices Performance



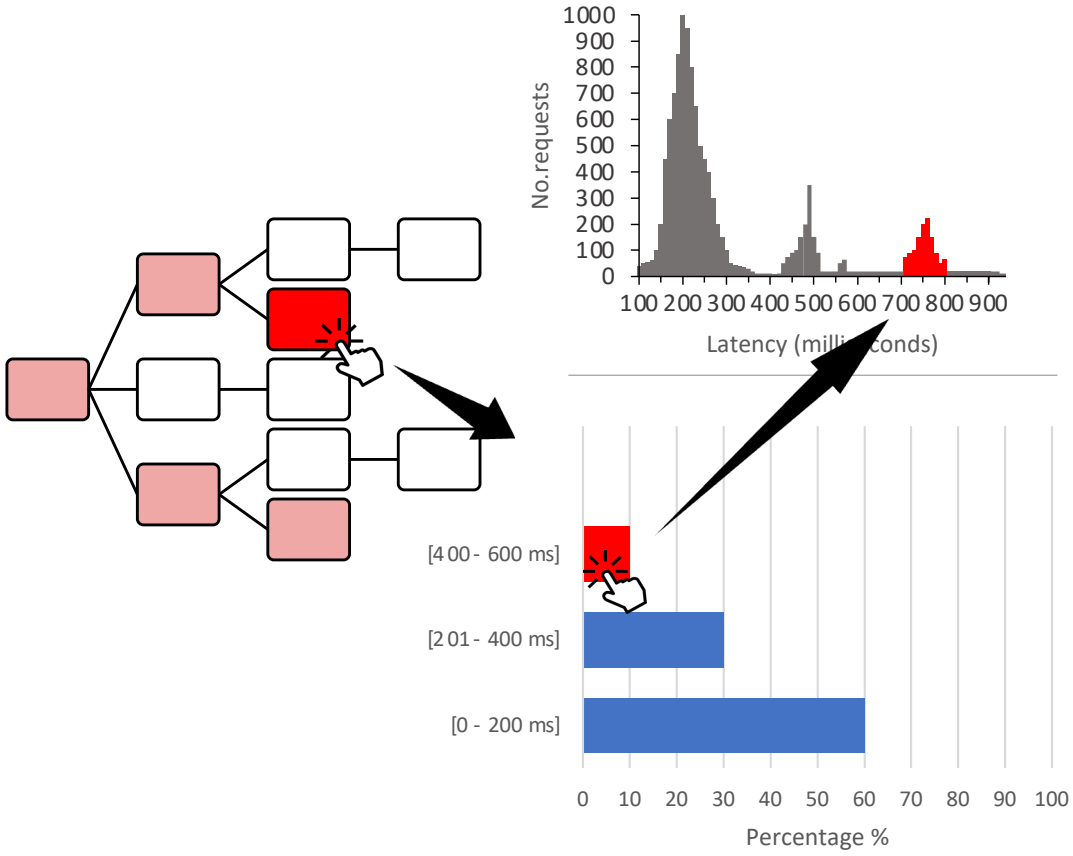
Interactive tree



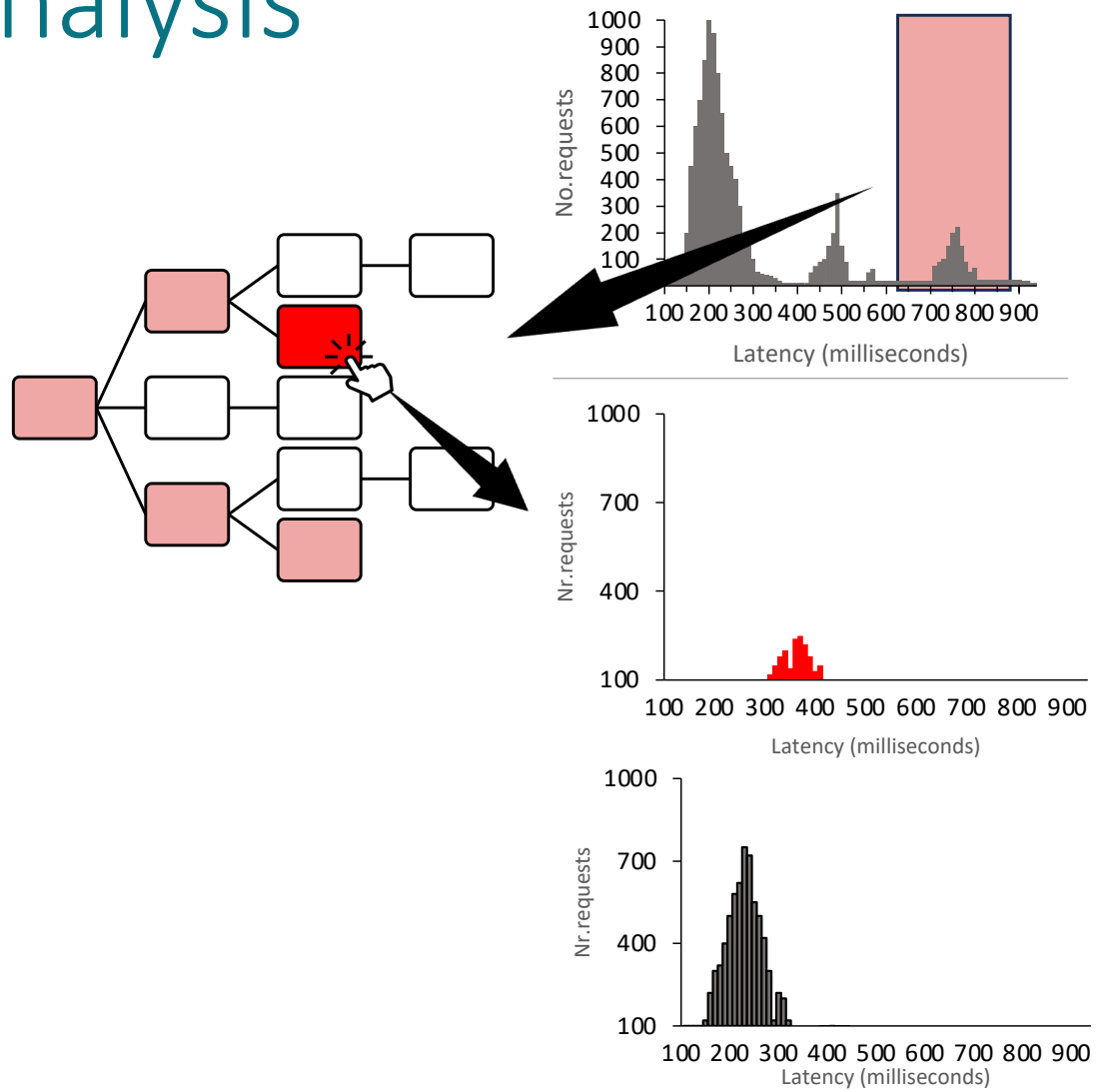
Histogram

Jessica Leone and Luca Traini. 2023. Enhancing Trace Visualizations for Microservices Performance Analysis. In Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal.

Forward Analysis



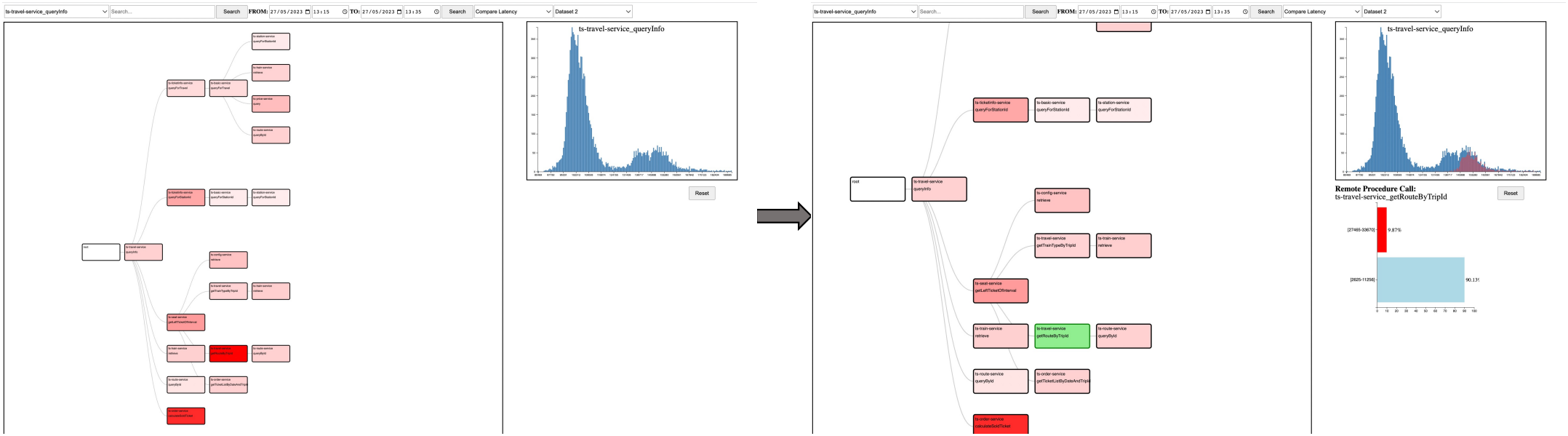
Backward Analysis



Evaluation

- Generation of 33 Datasets from a Benchmark Microservices System
- Injection of Synthetic Performance Anomalies into the System
- Empirical Evaluation Assessing the Effectiveness of VAMP in Performance Analysis

Results: Forward Analysis

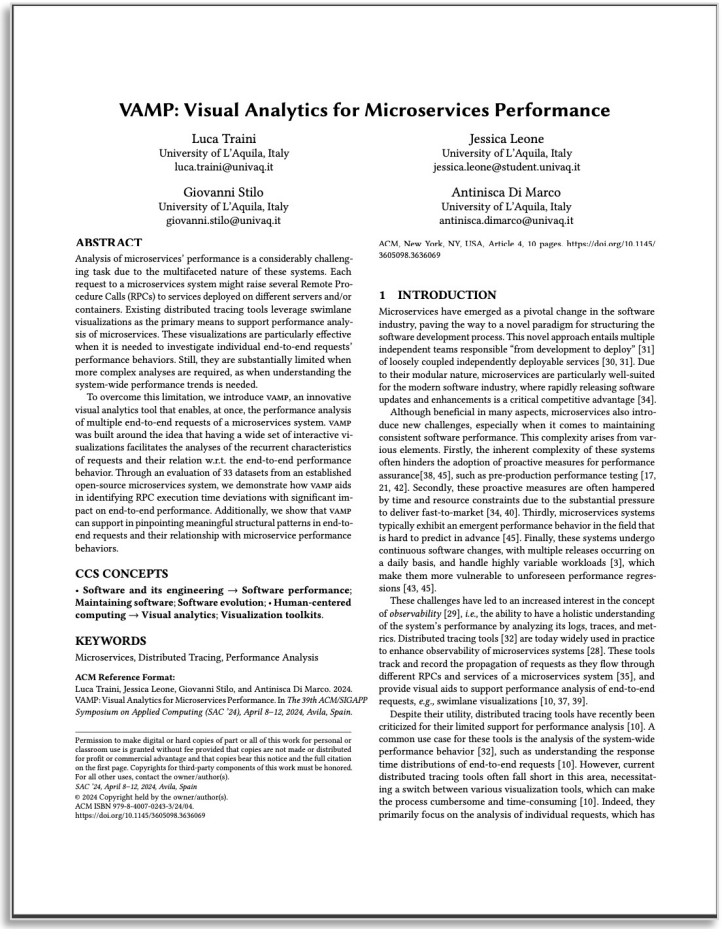


Conclusion

- Visual Analytics to Support Performance Analysis of Software Systems
- Two Interactive Components to Bridge the Gap Between Fine-Grained and Coarse-Grained Telemetry Data
- Evaluation on a Benchmark Microservices-Based System

VAMP: Visual Analytics for Microservices Performance

To appear:
In The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24)



VAMP: Visual Analytics for Microservices Performance

Luca Traini
University of L'Aquila, Italy
luca.traini@univaq.it

Jessica Leone
University of L'Aquila, Italy
jessica.leone@student.univaq.it

Giovanni Stilo
University of L'Aquila, Italy
giovanni.stilo@univaq.it

Antinisca Di Marco
University of L'Aquila, Italy
antinisca.dimarco@univaq.it

ABSTRACT

Analysis of microservices' performance is a considerably challenging task due to the multifaceted nature of these systems. Each request to a microservices system might raise several Remote Procedure Calls (RPCs) to services deployed on different servers and/or containers. Existing distributed tracing tools leverage swimlane visualizations as the primary means to support performance analysis of microservices. These visualizations are particularly effective when it is needed to investigate individual end-to-end requests' performance behaviors. Still, they are substantially limited when more complex analyses are required, as when understanding the system-wide performance trends is needed.

To overcome this limitation, we introduce VAMP, an innovative visual analytics tool that enables, at once, the performance analysis of multiple end-to-end requests of a microservices system. VAMP was built around the idea that having a wide set of interactive visualizations facilitates the analyses of the recurrent characteristics of requests and their relation w.r.t. the end-to-end performance behavior. Through an evaluation of 33 datasets from an established open-source microservices system, we demonstrate how VAMP aids in identifying RPC execution time deviations with significant impact on end-to-end performance. Additionally, we show that VAMP can support in pinpointing meaningful structural patterns in end-to-end requests and their relationship with microservice performance behaviors.

CCS CONCEPTS

• **Software and its engineering** → **Software performance**; **Maintaining software**; **Software evolution**; • **Human-centered computing** → **Visual analytics**; **Visualization toolkits**.

KEYWORDS

Microservices, Distributed Tracing, Performance Analysis

ACM Reference Format:
Luca Traini, Jessica Leone, Giovanni Stilo, and Antinisca Di Marco. 2024. VAMP: Visual Analytics for Microservices Performance. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.*

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3636069

ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3605098.3636069

1 INTRODUCTION

Microservices have emerged as a pivotal change in the software industry, paving the way to a novel paradigm for structuring the software development process. This novel approach entails multiple independent teams responsible "from development to deploy" [31] of loosely coupled independently deployable services [30, 31]. Due to their modular nature, microservices are particularly well-suited for the modern software industry, where rapidly releasing software updates and enhancements is a critical competitive advantage [34].

Although beneficial in many aspects, microservices also introduce new challenges, especially when it comes to maintaining consistent software performance. This complexity arises from various elements. Firstly, the inherent complexity of these systems often hinders the adoption of proactive measures for performance assurance[38, 45], such as pre-production performance testing [17, 21, 42]. Secondly, these proactive measures are often hampered by time and resource constraints due to the substantial pressure to deliver fast-to-market [34, 40]. Thirdly, microservices systems typically exhibit an emergent performance behavior in the field that is hard to predict in advance [45]. Finally, these systems undergo continuous software changes, with multiple releases occurring on a daily basis, and handle highly variable workloads [3], which make them more vulnerable to unforeseen performance regressions [43, 45].

These challenges have led to an increased interest in the concept of *observability* [29], *i.e.*, the ability to have a holistic understanding of the system's performance by analyzing its logs, traces, and metrics. Distributed tracing tools [32] are today widely used in practice to enhance observability of microservices systems [28]. These tools track and record the propagation of requests as they flow through different RPCs and services of a microservices system [35], and provide visual aids to support performance analysis of end-to-end requests, *e.g.*, swimlane visualizations [10, 37, 39].

Despite their utility, distributed tracing tools have recently been criticized for their limited support for performance analysis [10]. A common use case for these tools is the analysis of the system-wide performance behavior [32], such as understanding the response time distributions of end-to-end requests [10]. However, current distributed tracing tools often fall short in this area, necessitating a switch between various visualization tools, which can make the process cumbersome and time-consuming [10]. Indeed, they primarily focus on the analysis of individual requests, which has

Luca Traini, Jessica Leone, Giovanni Stilo, and Antinisca Di Marco. 2024. VAMP: Visual Analytics for Microservices Performance. In The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.