The 39th ACM/SIGAPP Symposium On Applied Computing (SAC 2024)
Software Architecture: Theory, Technology, and Applications (SA-TTA)

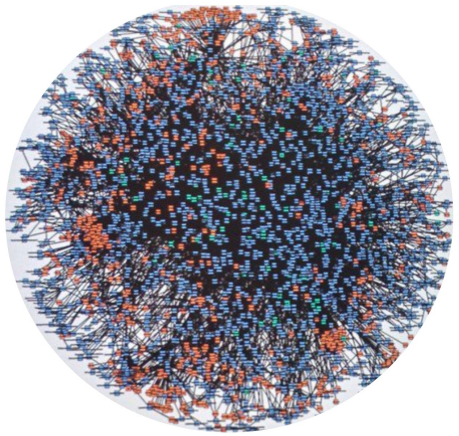
VAMP: Visual Analytics for Microservices Performance

Luca Traini, Jessica Leone, Giovanni Stilo, and Antinisca Di Marco
University of L'Aquila, Italy

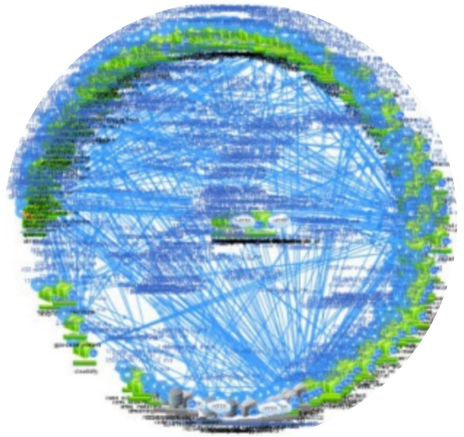
Microservices

- Loosely coupled independent services

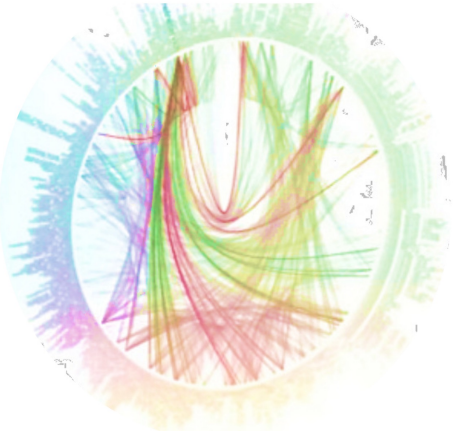
- Benefits:
  - Independent scaling
  - Independent development
  - Fast-paced release cycle



Amazon



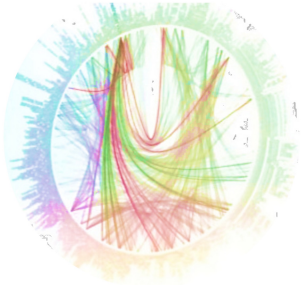
Netflix



Twitter

Performance Analysis of Microservices

- Complex interactions of multiple RPCs spread across multiple machines
- Frequent software releases can introduce performance issues
- Performance bugs can emerge from the interaction of multiple RPCs



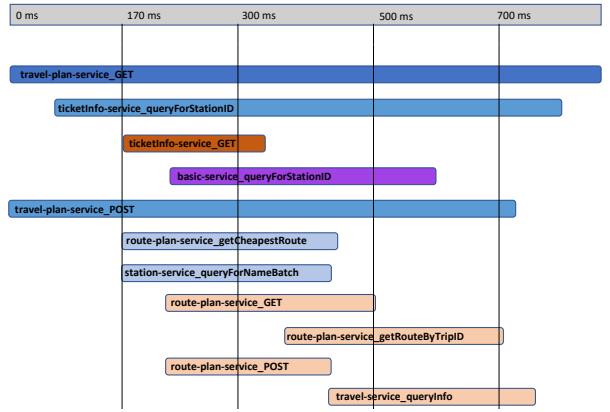
Twitter

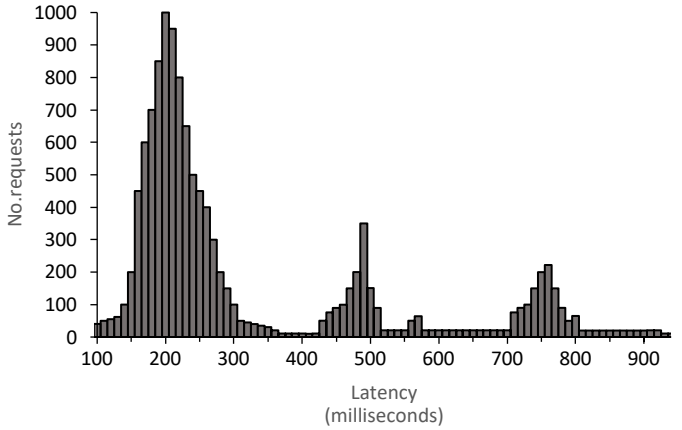
Distributed Tracing

- Capture the workflow of causally-related events (i.e., work done to process a request) within and among the components of a microservices system
- Swimlane visualizations as main visualization tool
- Often used in tandem with other visualization tools, such as Kibana

Distributed Tracing

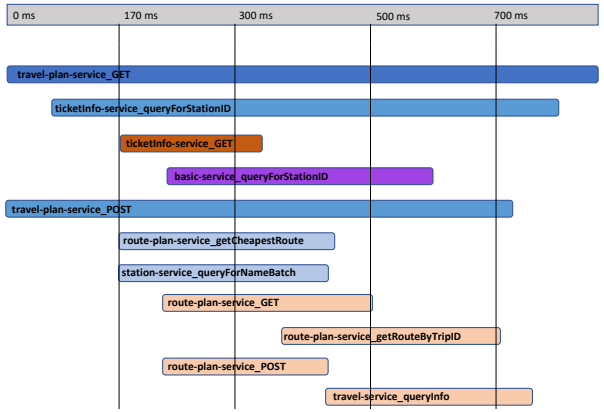
- Used for performance analysis of individual requests
- Individual request performance can be misleading without context
- Lack of support for aggregate analysis

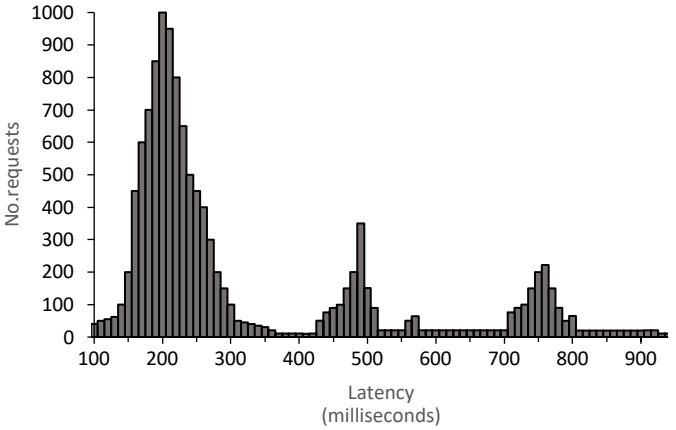




Objective

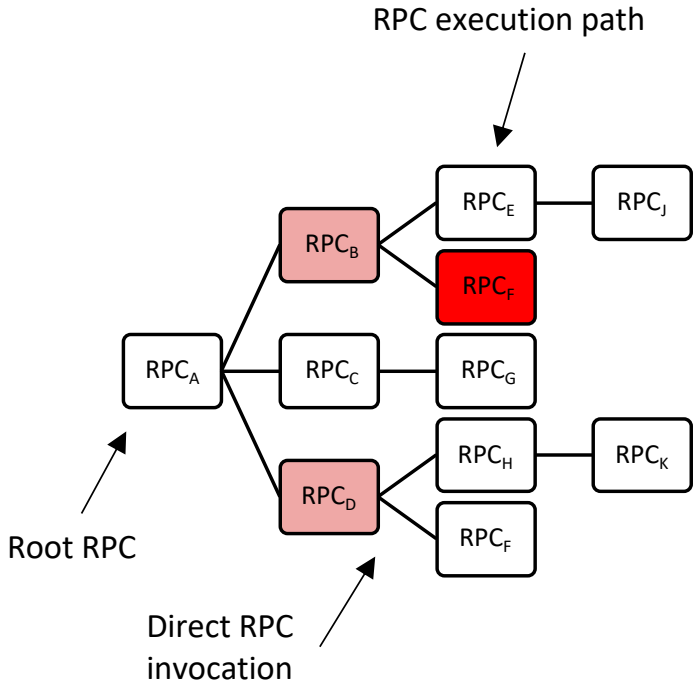
- Leverage visualization approaches to support performance analysis
- Highlights the relationship between request attributes and end-to-end latency
- Reduce the effort of switching between different tools





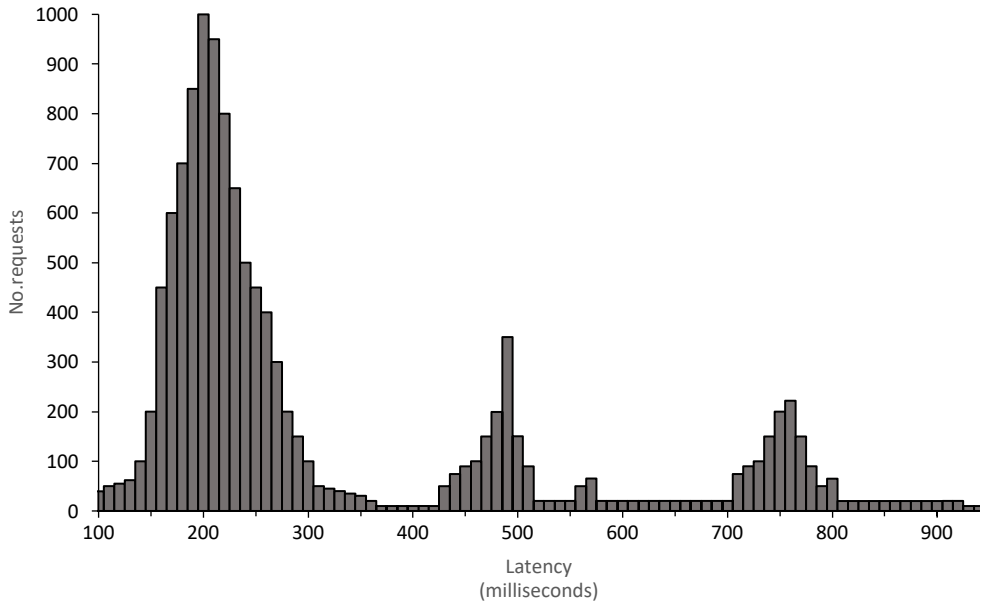
VAMP: Visual Analytics for Microservices Performance

Interactive tree



An aggregated view of the RPC workflows

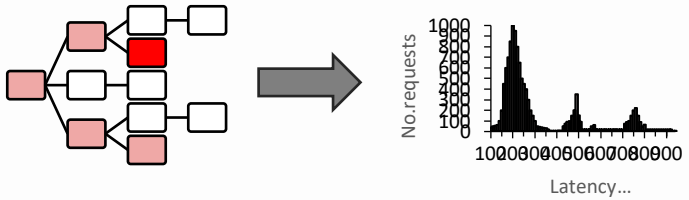
Histogram



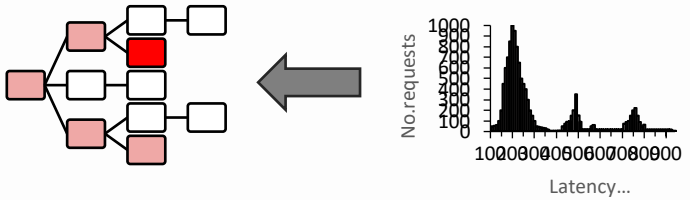
End-to-end latency distribution

Interaction Modalities

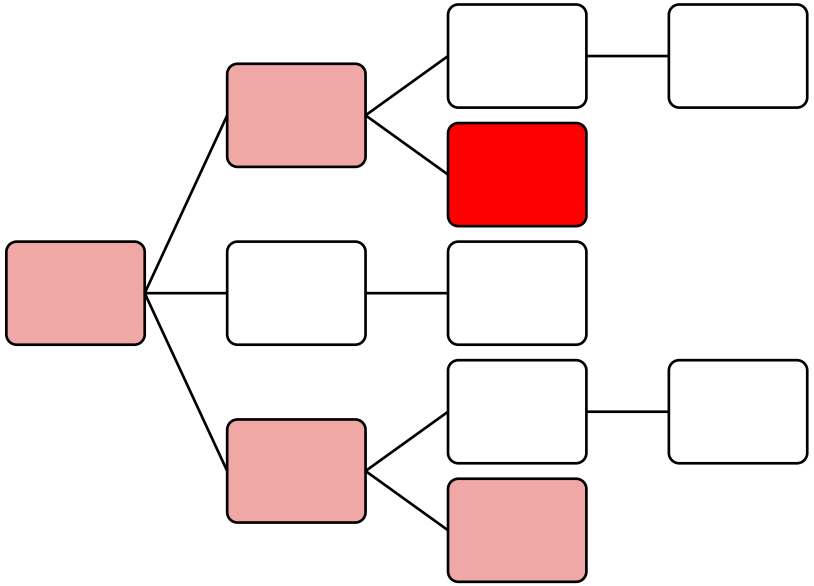
FORWARD ANALYSIS

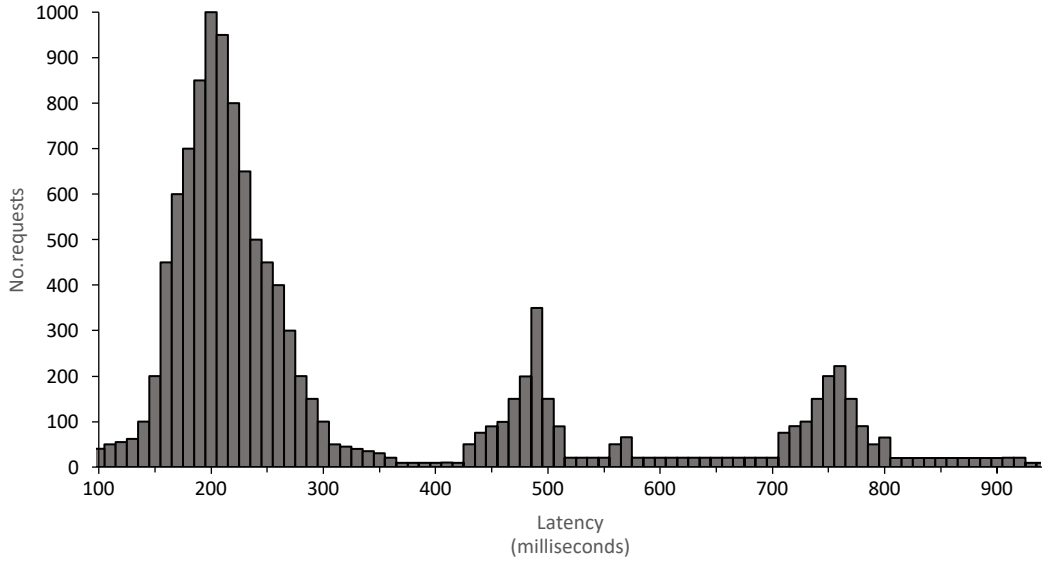


BACKWARD ANALYSIS



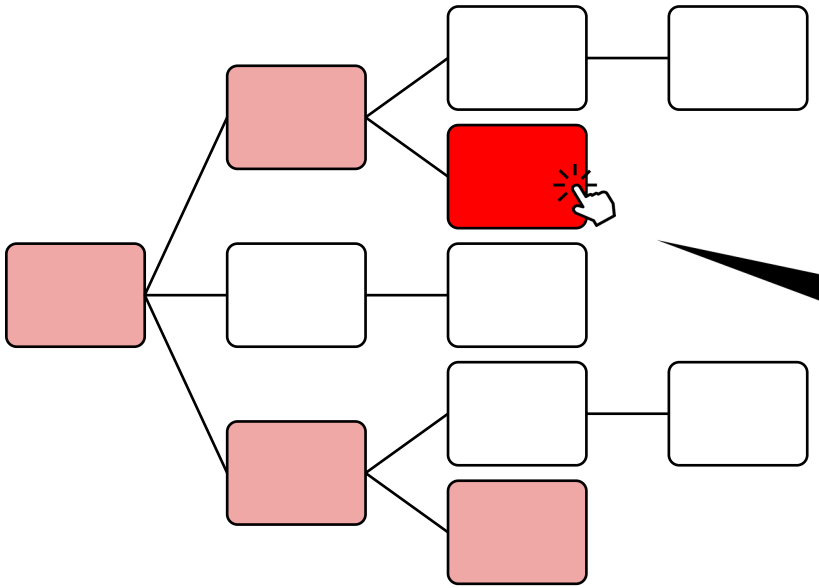
Forward Analysis

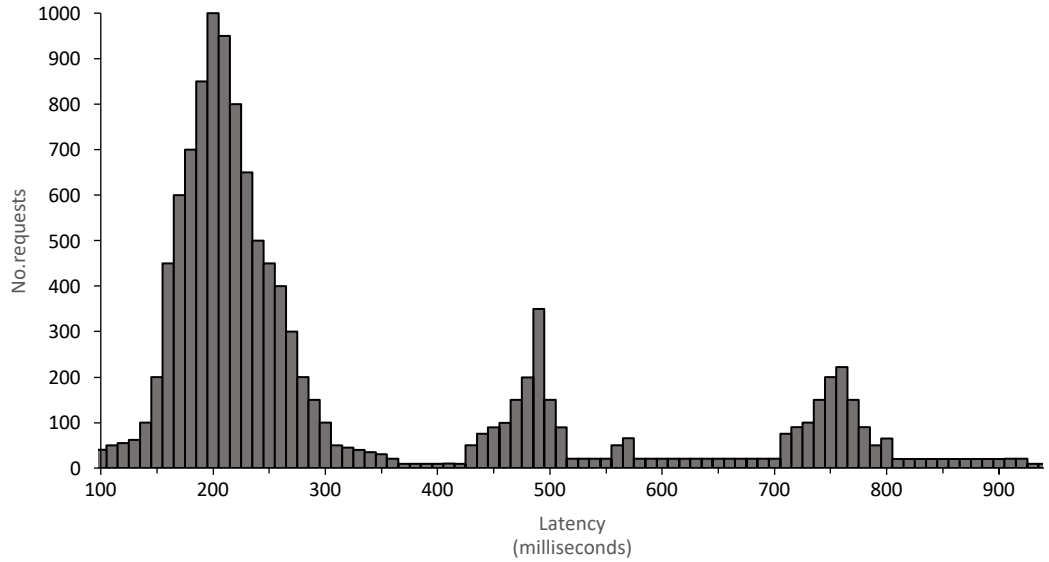


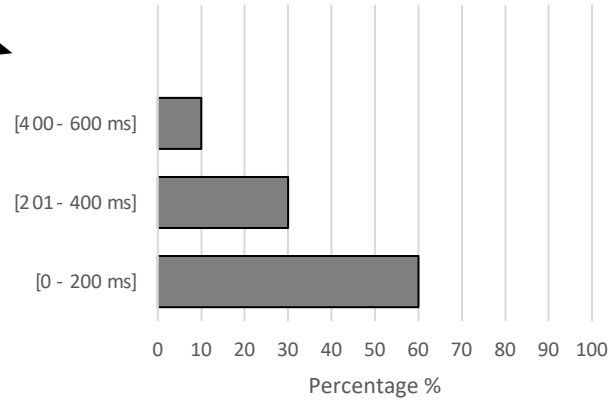


Color coding denotes variability in request attributes (e.g., execution time)

Forward Analysis

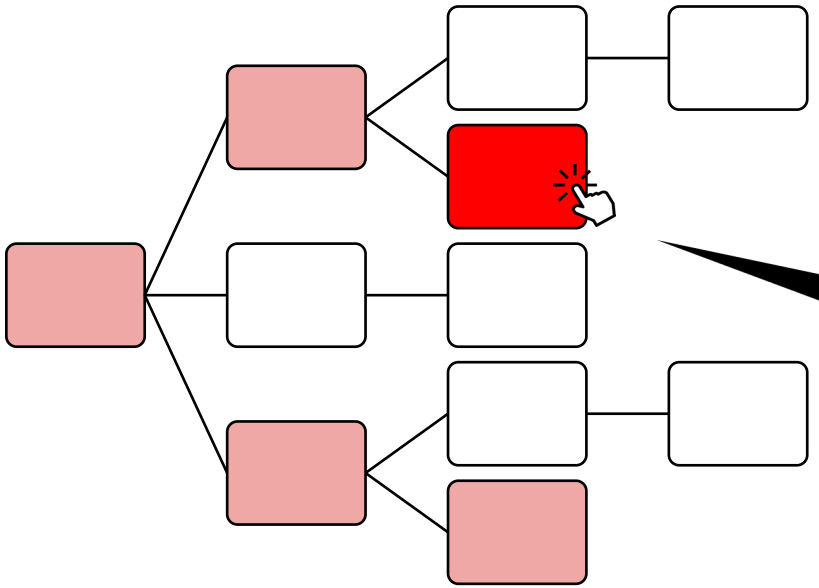


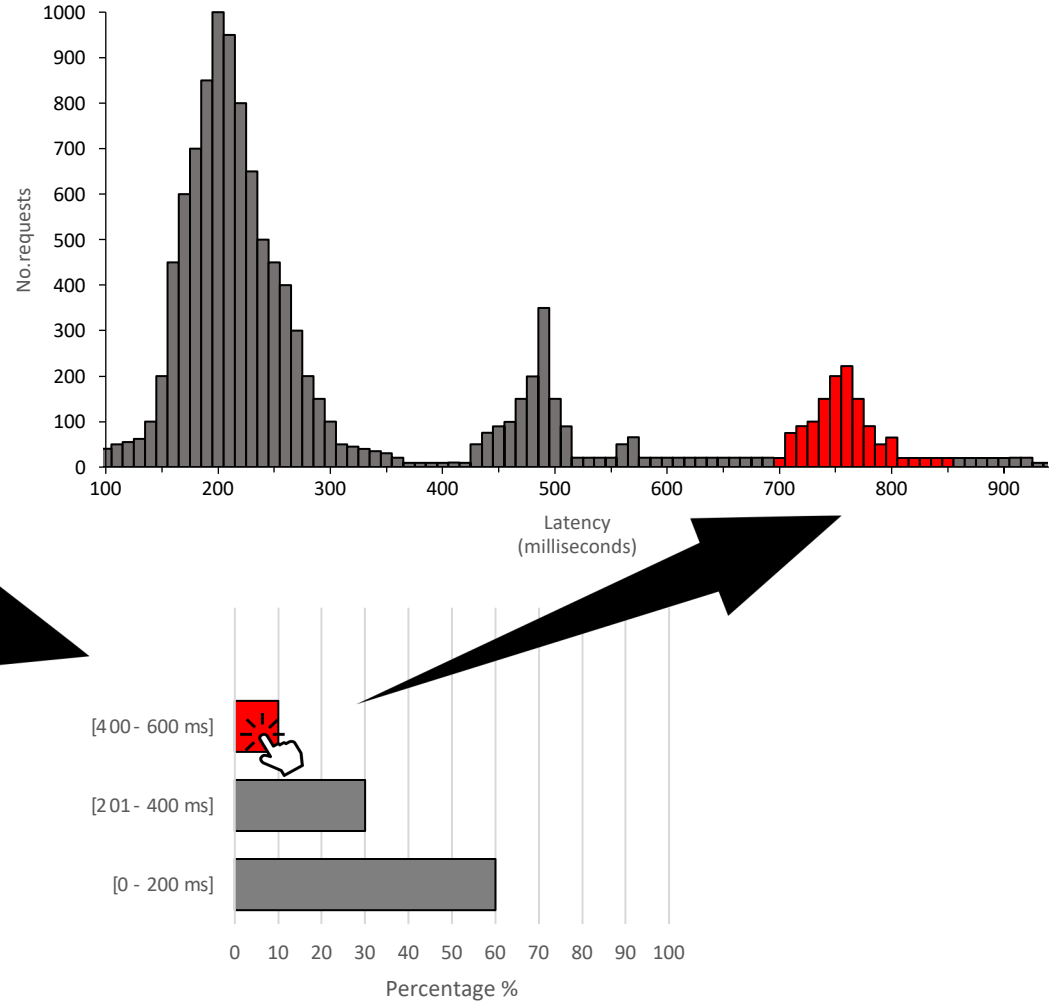




Clusters of recurrent execution time behavior computed using K-Means

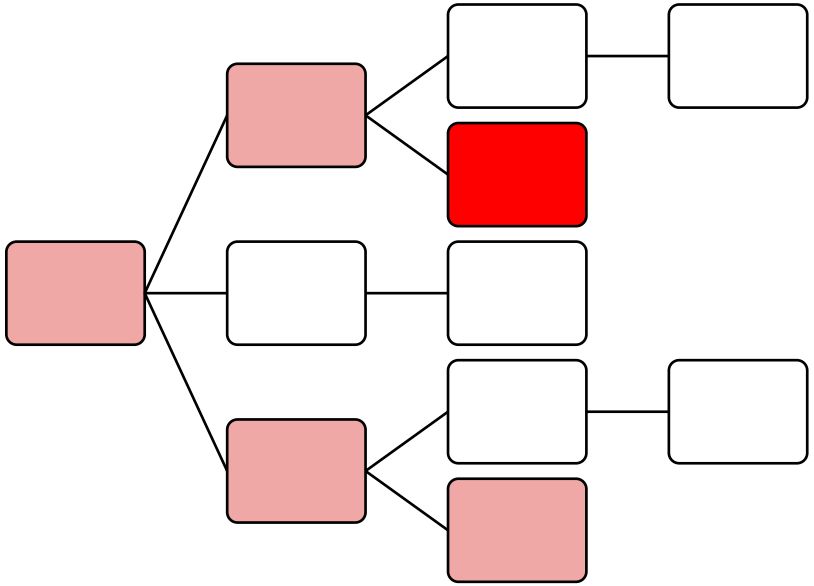
Forward Analysis

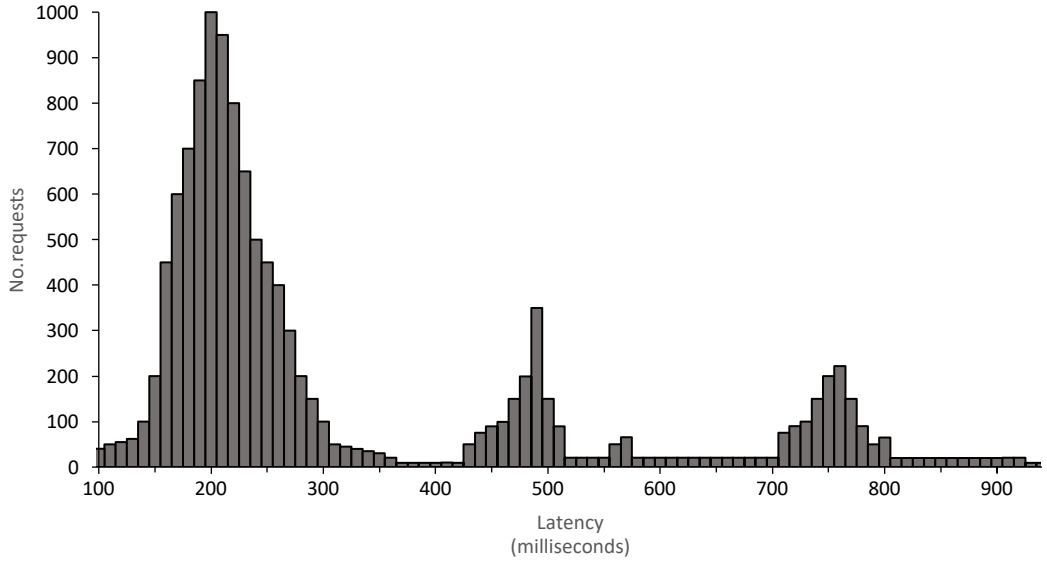




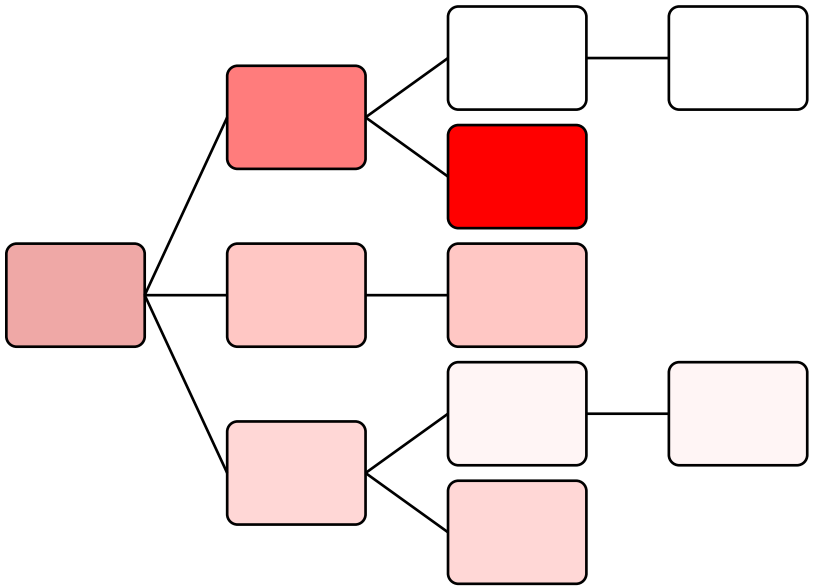
It shows how specific RPC executions time behaviors reflect in the end-to-end latency

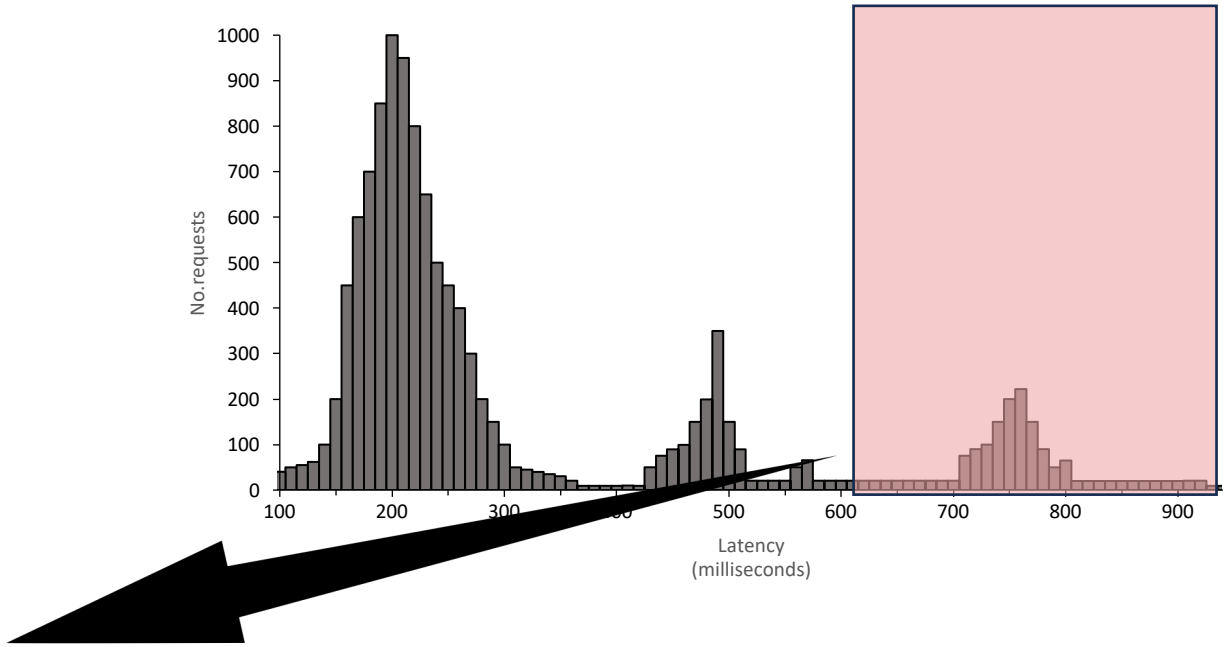
Backward Analysis





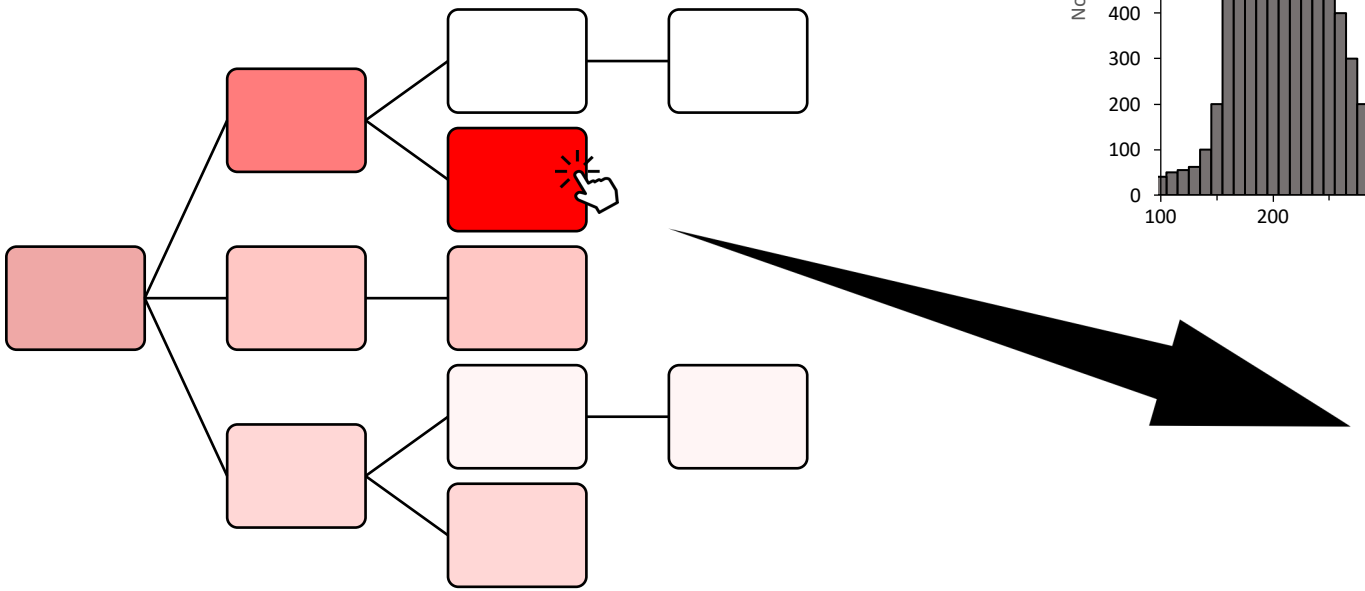
Backward Analysis

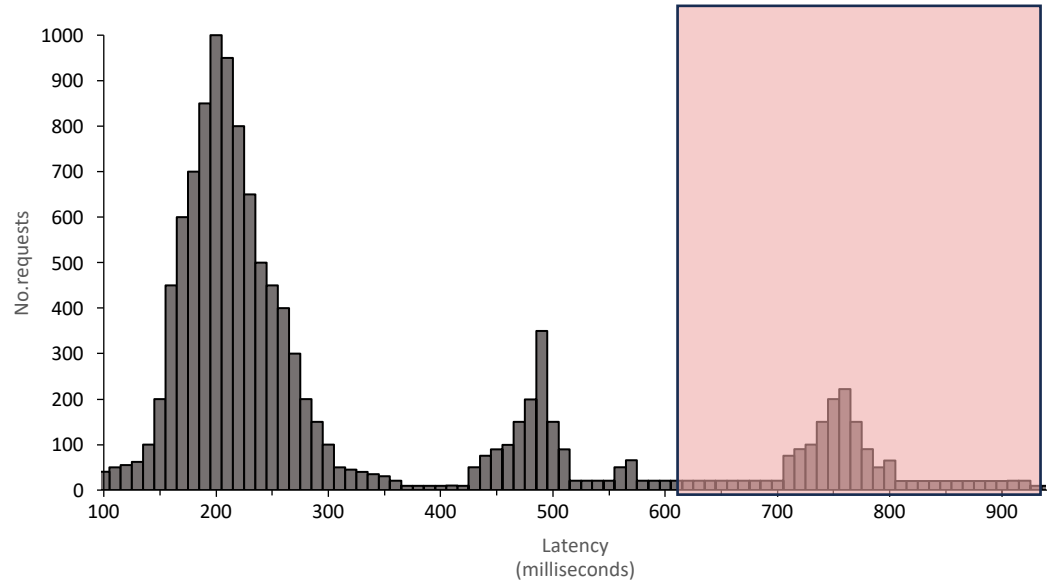


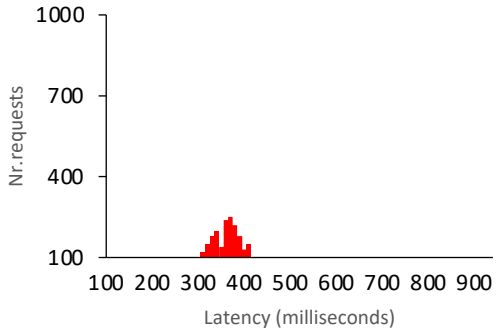


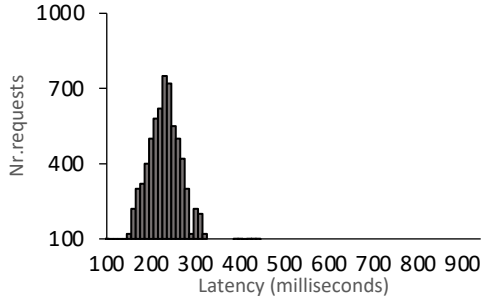
Color coding denotes divergence compared to other requests

Backward Analysis



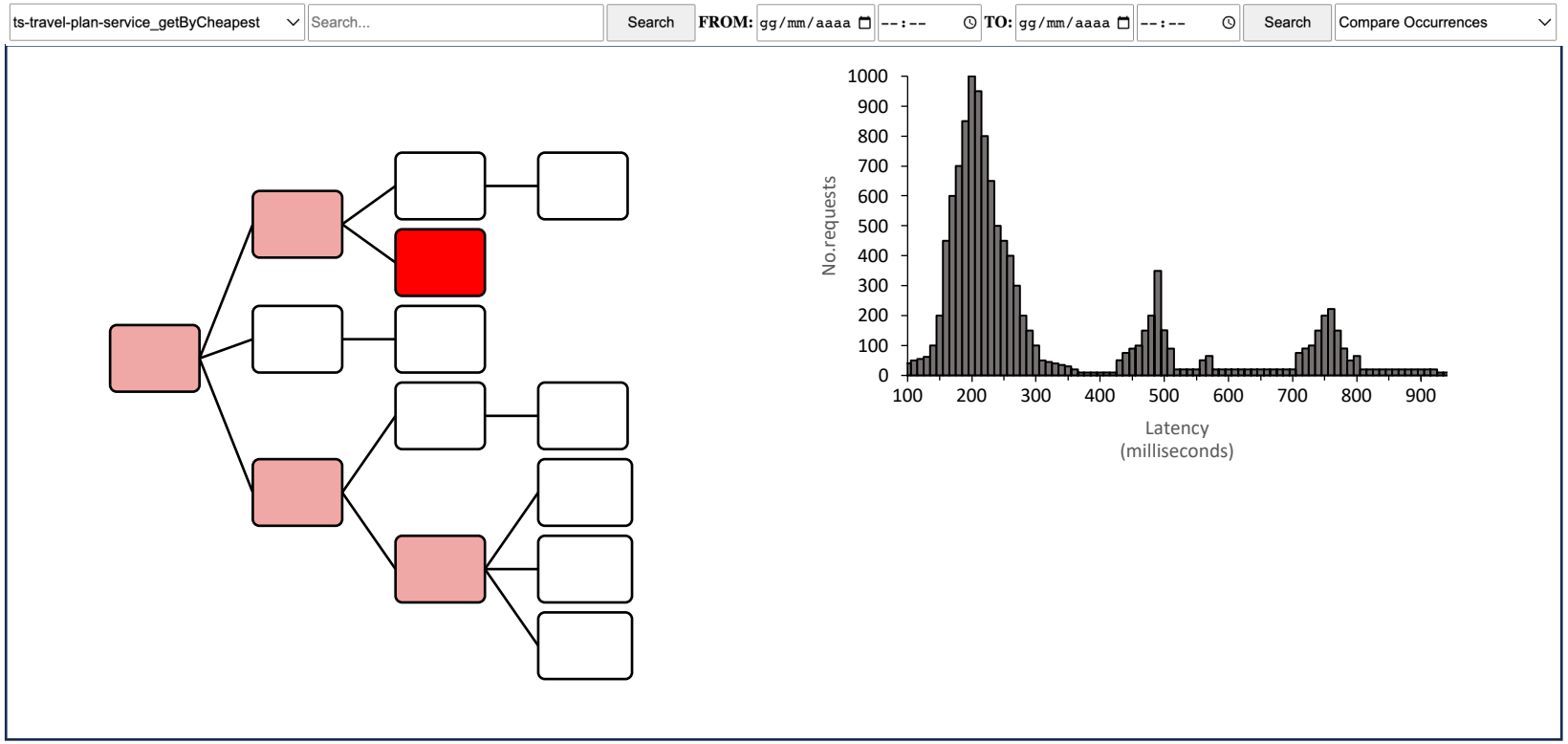






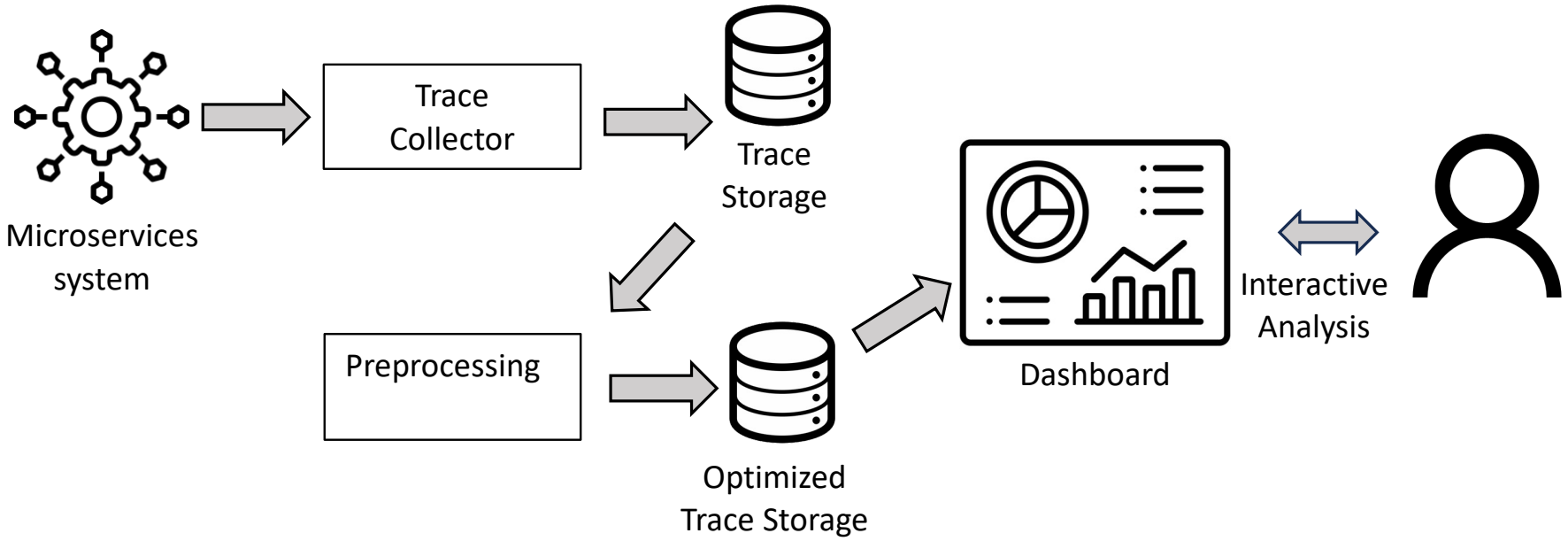
They shows the execution time distributions of the selected RPC

VAMP Dashboard

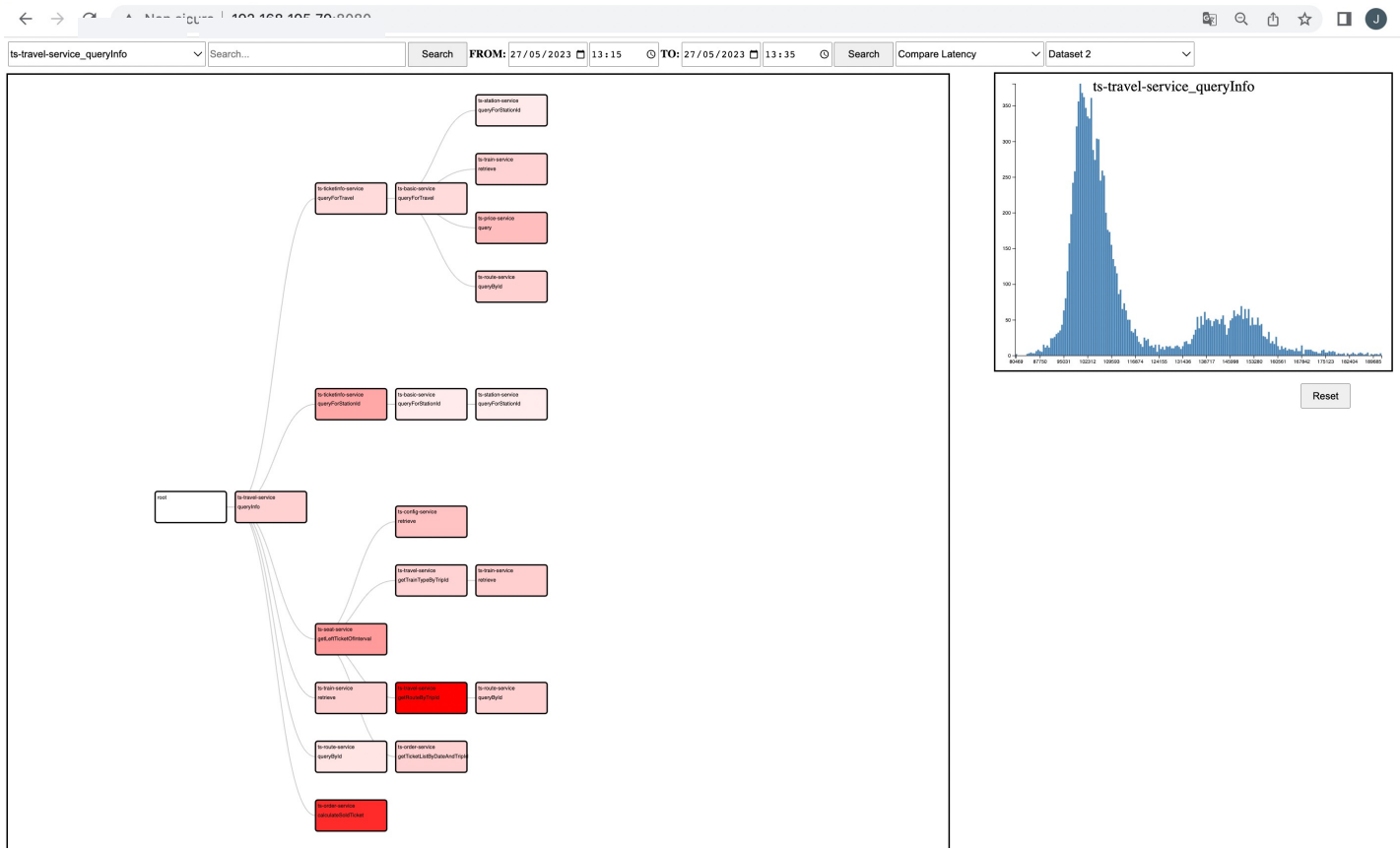


VAMP currently support two kinds of RPC attributes: *execution time* and *frequency of invocations*

VAMP's Workflow



Open-sourced implementation



https://github.com/lucatraini/VAMP

Evaluation

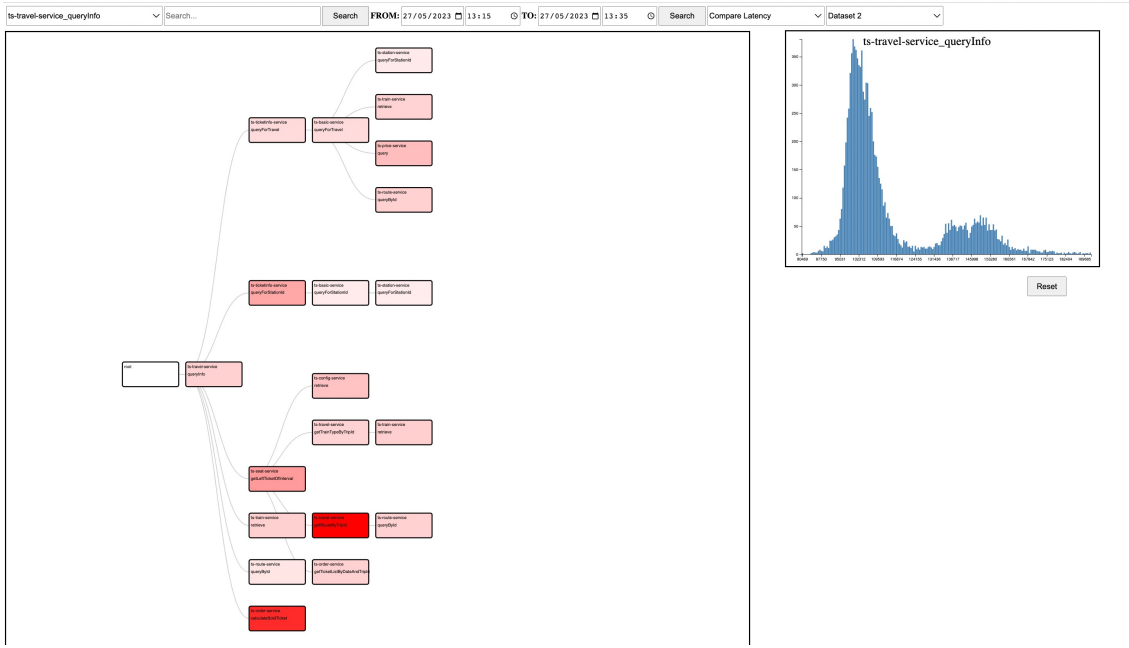
- Generation of 33 Datasets from a Benchmark Microservices System (i.e., Train-Ticket)
- Injection of Synthetic Performance Anomalies into the System
- Empirical Evaluation Assessing the Effectiveness of VAMP in Performance Analysis

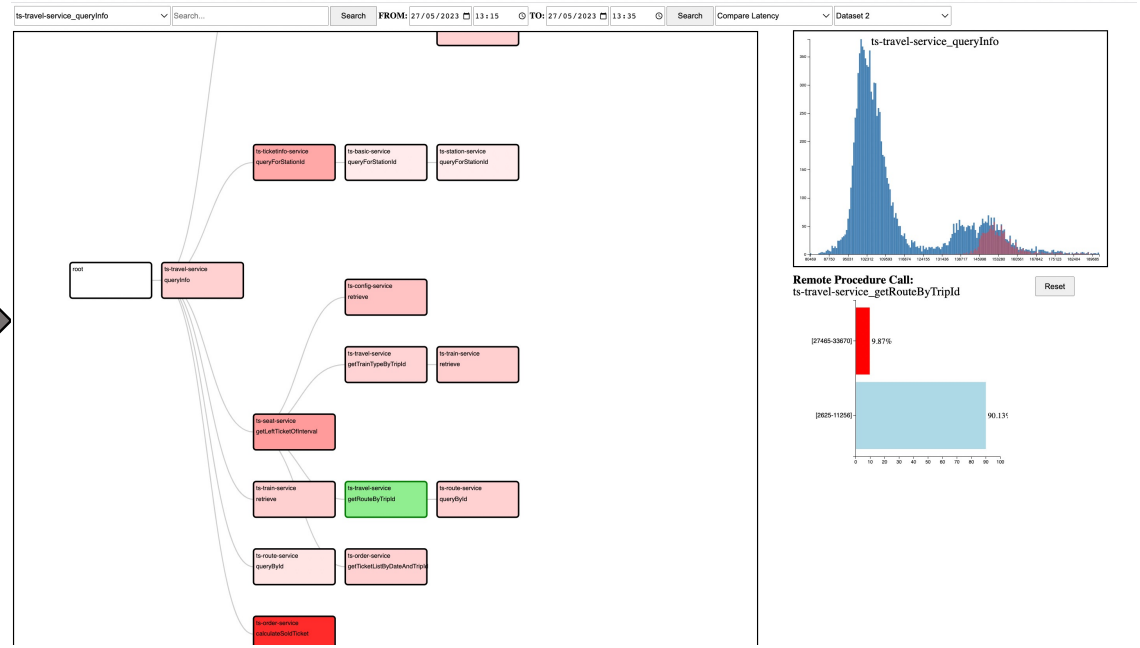
Results

- VAMP helps to rapidly identify RPCs affected by performance issues in 18 out of the 20 datasets involving performance issues

- VAMP aids in understanding how varying frequencies of RPCs influence end-to-end latency

Example


ts-travel-service_queryInfo
Search
FROM: 27/05/2023 13:15
TO: 27/05/2023 13:35
Search
Compare Latency
Dataset 2
ts-travel-service_queryInfo
Reset


ts-travel-service_queryInfo
Search
FROM: 27/05/2023 13:15
TO: 27/05/2023 13:35
Search
Compare Latency
Dataset 2
root
ts-travel-service
queryInfo
ts-ticketinfo-service
queryForStationId
ts-basic-service
queryForStationId
ts-station-service
queryForStationId
ts-config-service
retrieve
ts-travel-service
getTrainTypeByTripId
ts-train-service
retrieve
ts-seat-service
getLeftTicketOfInterval
ts-train-service
retrieve
ts-travel-service
getRouteByTripId
ts-route-service
queryById
ts-route-service
queryById
ts-order-service
getTicketListByDateAndTripId
ts-order-service
calculateSoldTicket
ts-travel-service_queryInfo
Reset
Remote Procedure Call:
ts-travel-service_getRouteByTripId
[27465-33870]
9.87%
[2625-11256]
90.13%

Conclusion and Future work

- VAMP is a novel visual analytics tool for microservices performance analysis
- VAMP can be effectively used to understand the relationship between the RPC attributes and end-to-end response time
- We plan to enhance the efficiency of our tool to facilitate its transition to practice