Comprehensive Latency issue Diagnosis in Microservices Using Enhanced Spectrum Analysis

Maryam Ekhlasi

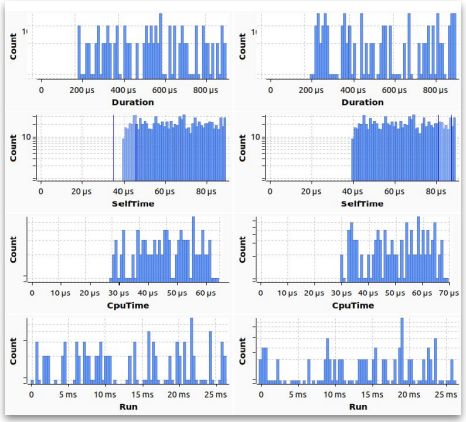
May. 30th, 2024

Polytechnique Montreal

DORSAL Laboratory

Motivations

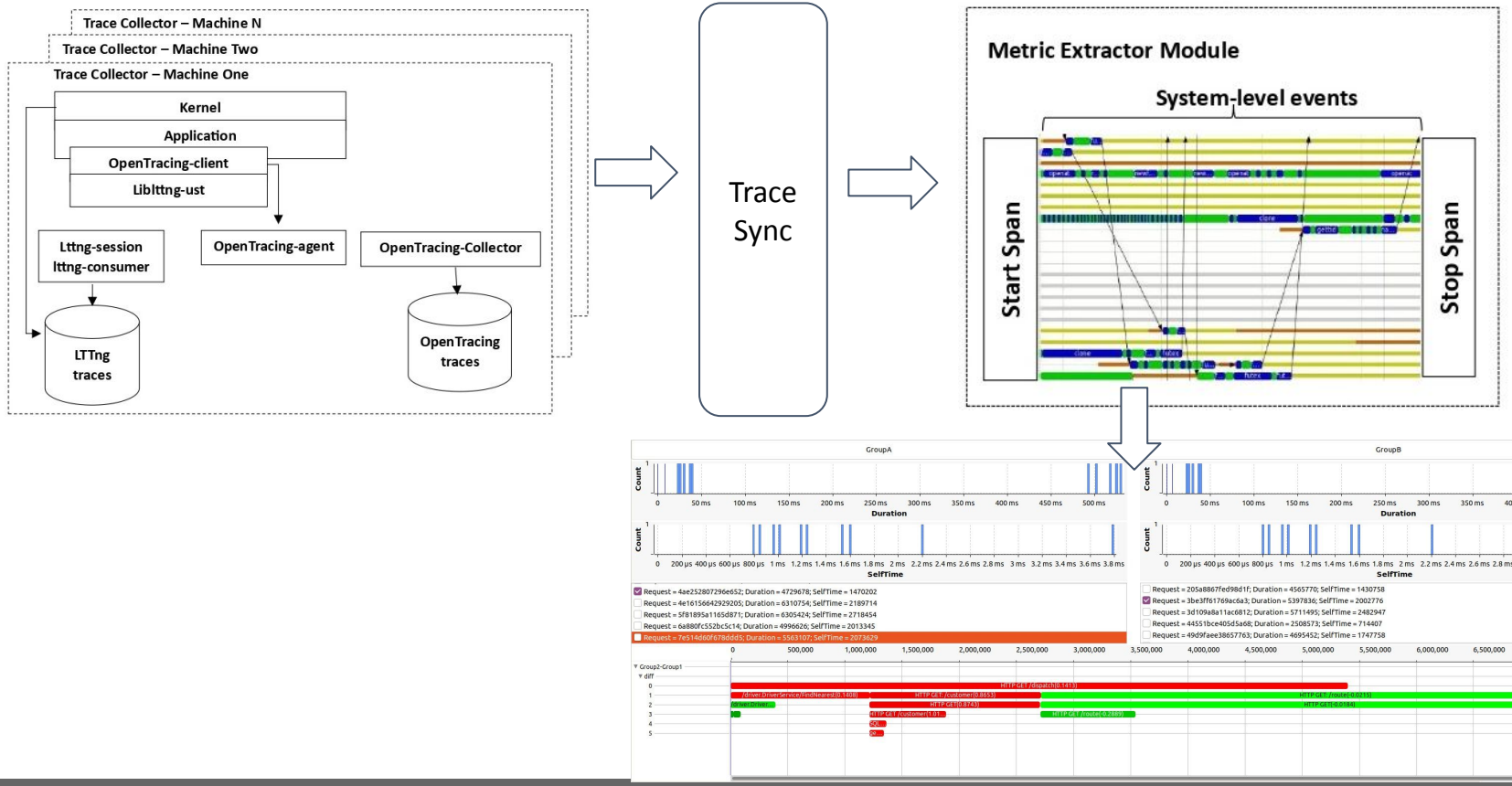
- Manual analysis of distributed requests is inefficient, requires expert knowledge
- Enhance root cause localization in Microservice systems
- Improve the existing request-centric view



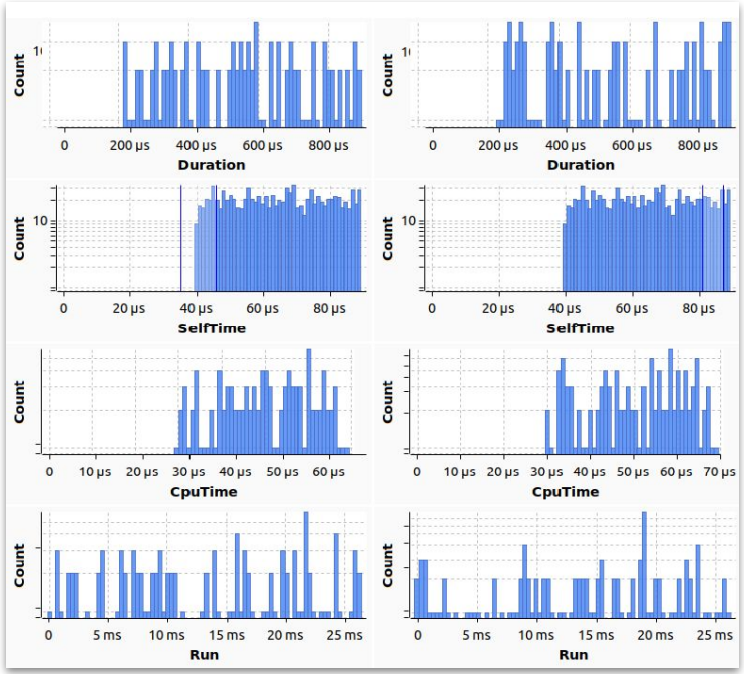
Our Goal

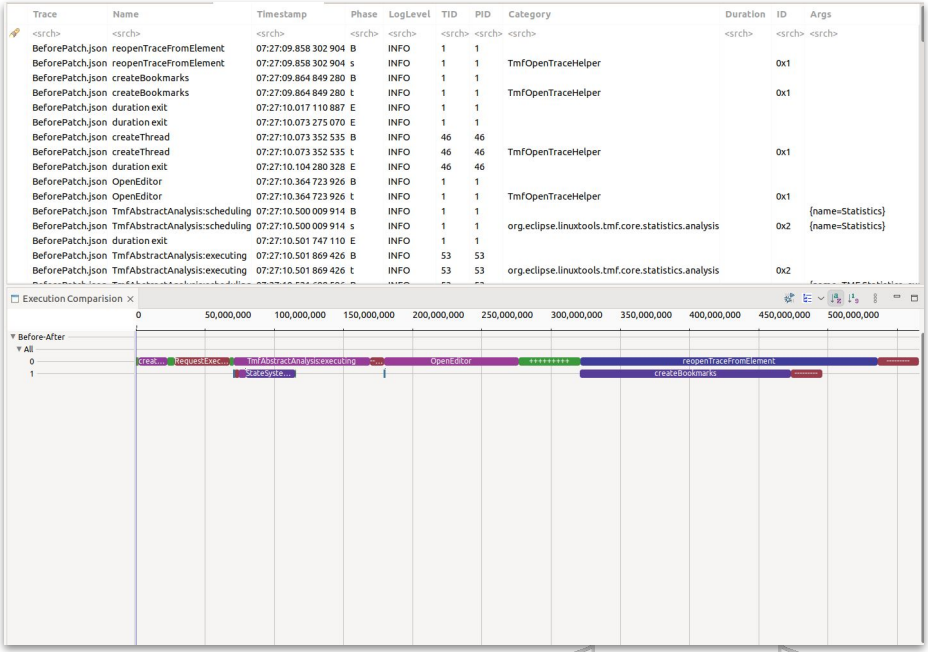
1. Locating the performance problem through a distributed system. (Accomplished)
2. Comparing two sets of executions to evaluate the differences in terms of performance. (Accomplished)
3. Providing sets of views to highlight the differences and speed up problem diagnosis. (Accomplished)
4. Grouping similar requests, with closely related but not identical structure. (Ongoing)
5. Finding the normal execution threshold for each group. (Ongoing)

Architecture



Filtering Module





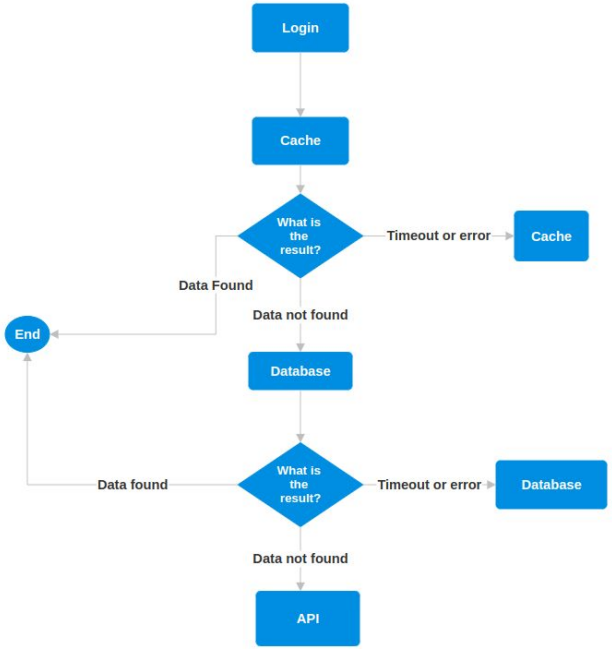
Real world Problem

Engineers want to understand occasional delays in their login processes. The login sequence typically follows these steps:

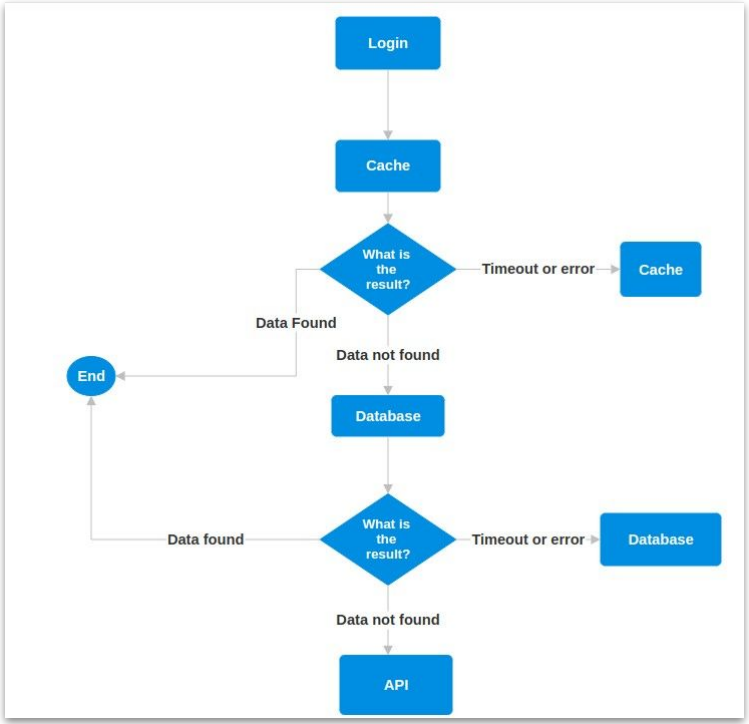
- Request information from the cache.
- Retry three times if the cache fails.
- Request data from the database if cache retrieval fails.
- Retry from an alternative database up to three times

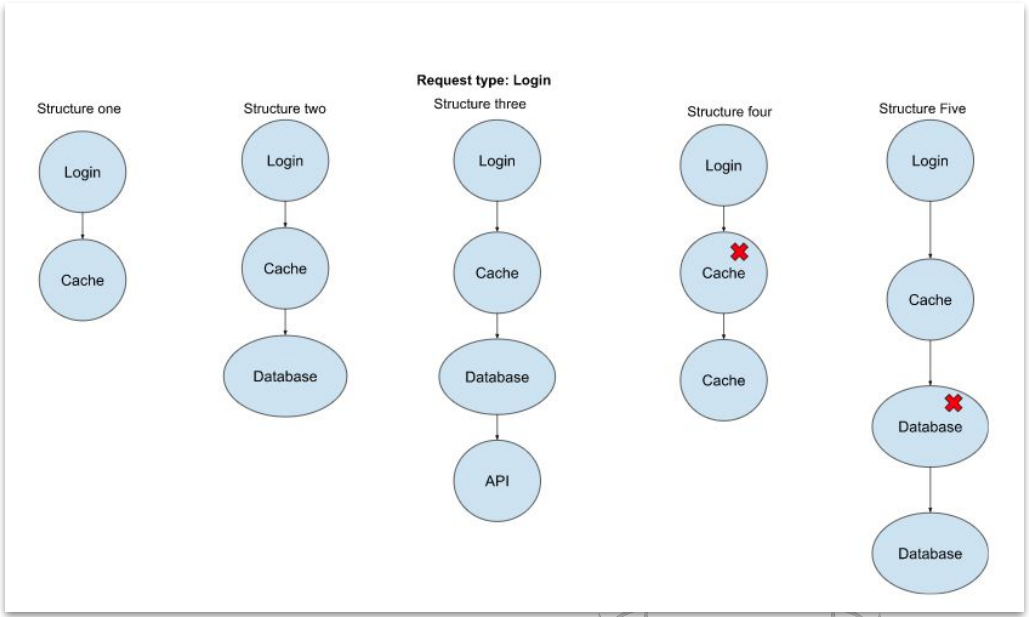
if the main database fails

- Query an external API if the secondary database fails.

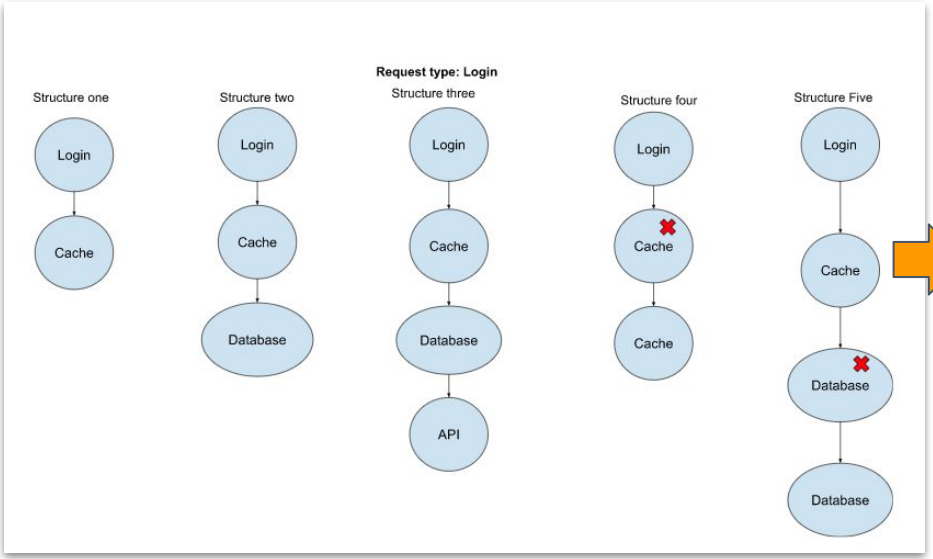


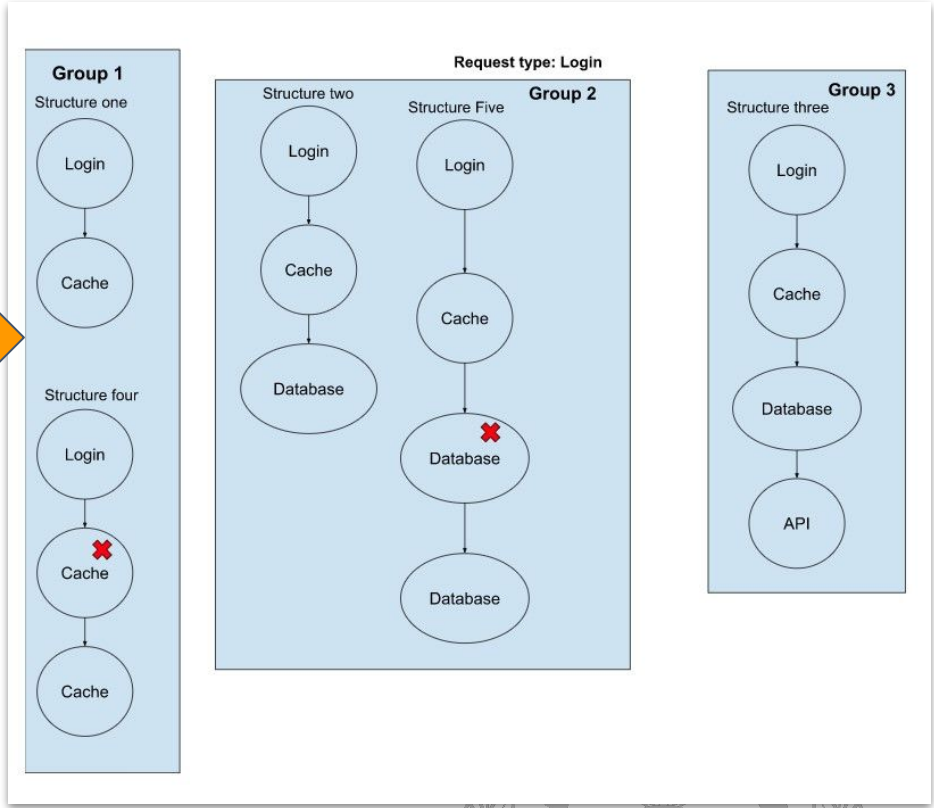
Example


Login
Cache
What is the result?
Timeout or error
Cache
Data Found
Data not found
End
Database
What is the result?
Data found
Timeout or error
Database
Data not found
API


Request type: Login
Structure one
Login
Cache
Structure two
Login
Cache
Database
Structure three
Login
Cache
Database
API
Structure four
Login
Cache
Cache
Structure Five
Login
Cache
Database
Database

Different Structures





Concerns

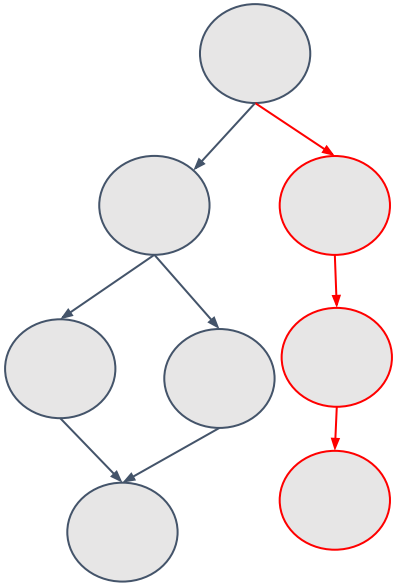
1. One simple request may have several different structures.
2. Graph/tree mining approaches are computationally expensive.
3. Parallel services (branches) can be challenging.

Strategy

1. Weighting requests based on their ability to localize root causes of performance problems.
2. Pay more attention to less frequent request types to ensure a balanced distribution of different request occurrences.
3. Using MicroRank, to identify service instances that may contribute to latency issues.

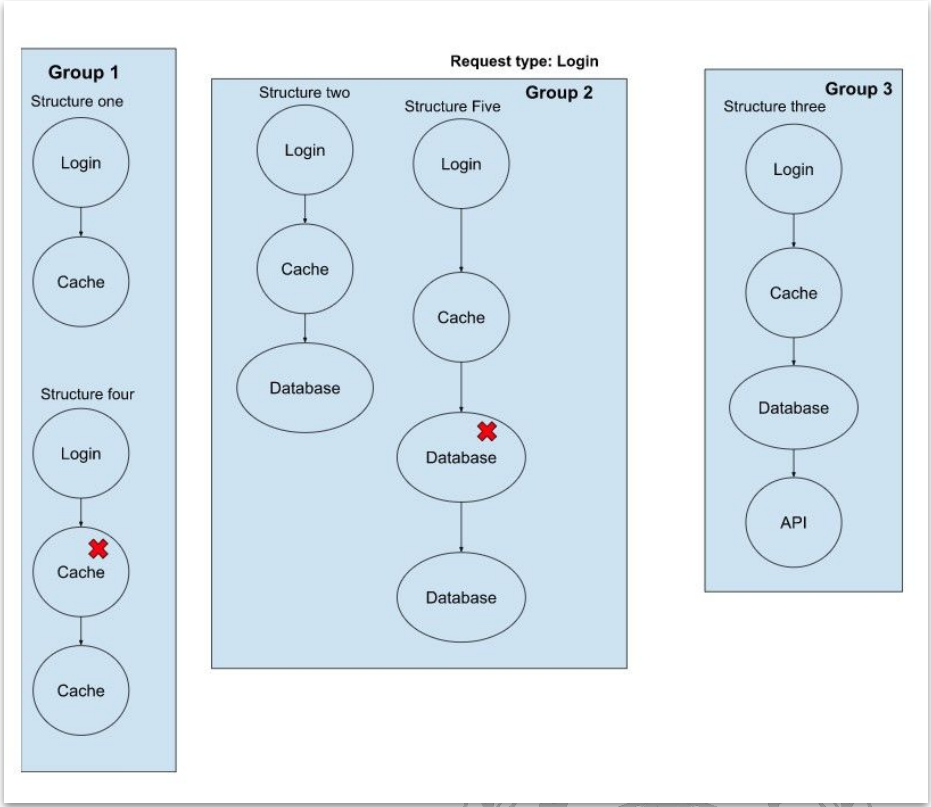
Strategy

1. Grouping requests based on their types
2. Extracting the critical path of each request



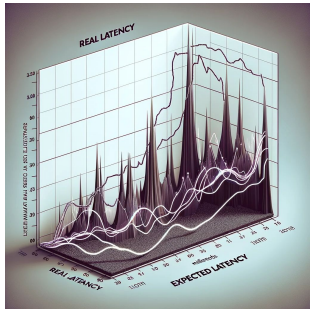
Strategy

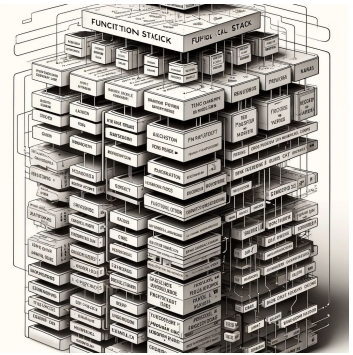
- Aggregation of Identical service calls
- Group the similar sequences of service calls



Strategy

- Evaluating the Expected Latency
- Comparing the real latency and expected latency
- Personalised PageRank Algorithm
- Weighted Spectrum Ranker to list the order of suspicious services
- Extract the call stack of the functions within each service for root cause analysis





Refrences

- Guangba Yu, Pengfei Chen, Hongyang Chen, Zijie Guan, Zicheng Huang, Linxiao Jing, Tianjun Weng, Xinmeng Sun, and Xiaoyun Li. 2021. MicroRank: End-to-End Latency Issue Localization with Extended Spectrum Analysis in Microservice Environments. In Proceedings of the Web Conference 2021 (WWW '21). Association for Computing Machinery, New York, NY, USA, 3087–3098. https://doi.org/10.1145/3442381.3449905

Thank you

Email: maryam.ekhlasi@polymtl.ca