Performance Issues Detection by Comparing Nested Execution Traces in Distributed Systems

Maryam Ekhlasi

January 28th, 2022

Polytechnique Montreal

DORSAL Laboratory

Motivation

- Help users understand what is happening when something is different
  - Performance differences between versions of the application.
  - Performance differences between system configurations.
  - Why some requests are slower than others?

Previous Work at Dorsal Lab

electronics

MDPI

Article

Combining Distributed and Kernel Tracing for Performance Analysis of Cloud Applications

Loïc Gelle [1], Naser Ezzati-Jivan [2,*] and Michel R. Dagenais [1]

[1] Computer Engineering and Software Engineering Department, Ecole Polytechnique Montreal, Montreal, QC H3T 1J4, Canada; loic.gelle@polymtl.ca (L.G.); michel.dagenais@polymtl.ca (M.R.D.)
[2] Computer Science Department, Brock University, St. Catharines, ON L22 3A1, Canada
* Correspondence: nezzati@brocku.ca

462 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 28, NO. 2, FEBRUARY 2017

Diagnosing Performance Variations by Comparing Multi-Level Execution Traces

François Doray, *Student Member, IEEE* and Michel Dagenais, *Senior Member, IEEE*

Roadmap

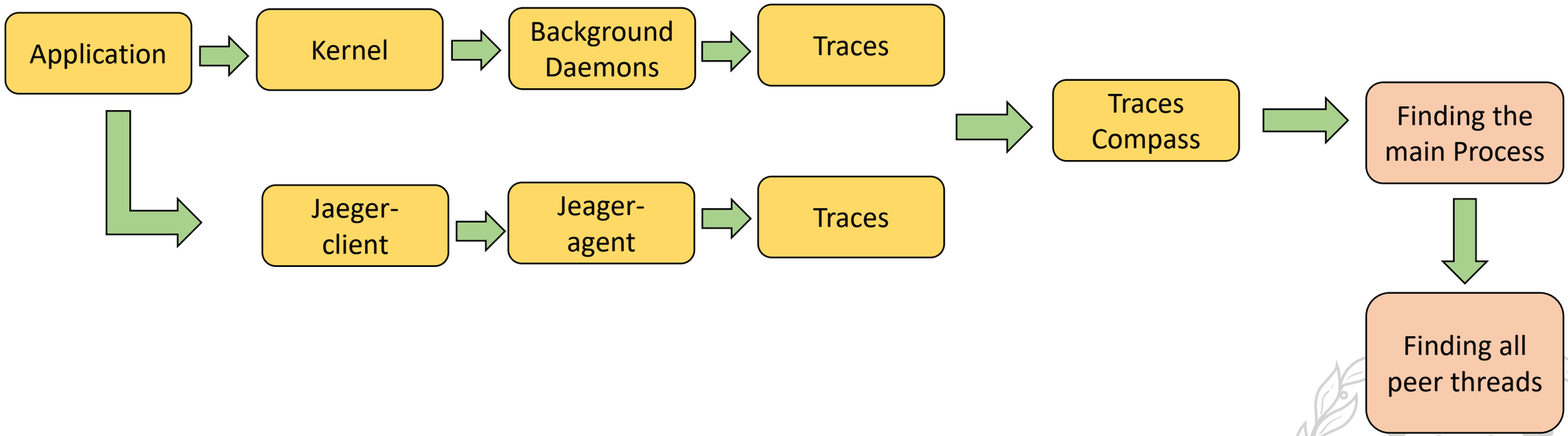
- Detecting nested trace requests.
- Collecting User-space and kernel-level events of all machines.
- Comparing similar traces for detecting performance issues.
- Extend Trace Compass or one of the performance analysis tools to analyze micro-services.

Strategy

- Creating trees of several threads from the nested requests.
- Linking system calls, critical path, and remote procedure calls of all related threads.
- Grouping and showing the differences between similar execution traces.

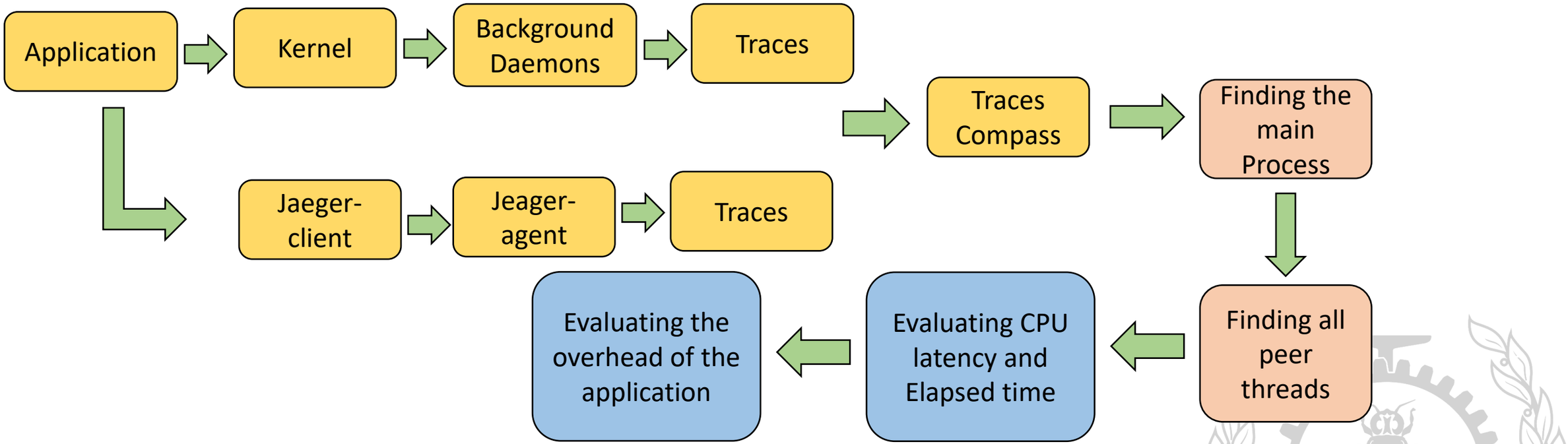
Accomplished

- Instrumenting Jaeger python source code with LTTng
  - Defining the start and endpoint of each span.
- Illustrating the communication between the main thread and its peer on the kernel side.



Accomplished

- Calculating the CPU latency and elapsed time for all related spans of the specific request through distributed systems.
- Evaluating my work by applying workload to the system.



Future Goal

- Mapping the execution flow of similar requests.
- Grouping similar traces and finding differences.

Thank you

Email: maryam.ekhlasi@polymtl.ca

Source Code: https://github.com/maryamekhlasi/jaeger-client-python.git