Performance Issues Detection by Comparing Nested Execution Traces in Distributed Systems

Maryam Ekhlasi

May 16th, 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

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 47, NO. 2, FEBRUARY 2021 243

Fault Analysis and Debugging of Microservice Systems: Industrial Survey, Benchmark System, and Empirical Study

Xiang Zhou, Xin Peng, Tao Xie, *Fellow, IEEE*, Jun Sun, Chao Ji, Wenhai Li, and Dan Ding

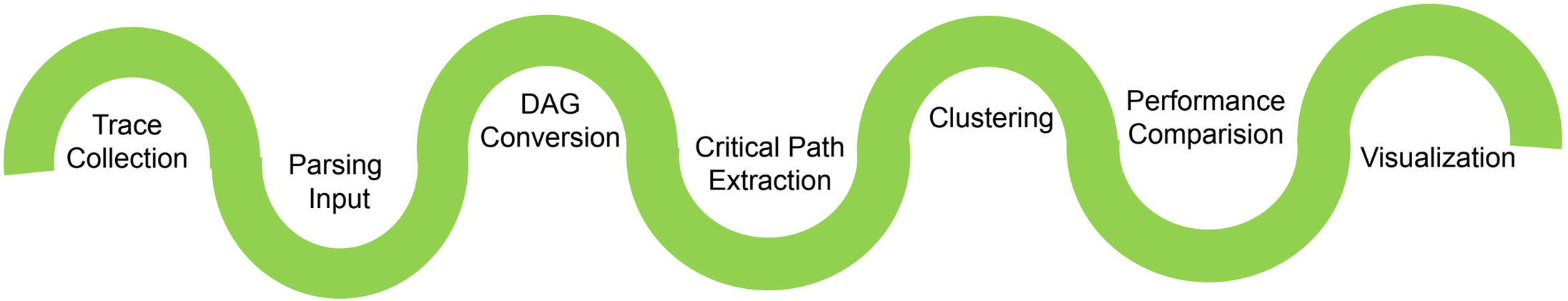
2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM)

DepGraph: Localizing Performance Bottlenecks in Multi-Core Applications Using Waiting Dependency Graphs and Software Tracing

Naser Ezzati-Jivan, Brock University, nezzatijivan@brocku.ca
Quentin Fournier, Polytechnique Montreal, quentin.fournier@polymtl.ca
Michel R. Dagenais, Polytechnique Montreal, michel.dagenais@polymtl.ca
Abdelwahab Hamou-Lhadj, Concordia University, wahab.hamou-lhadj@concordia.ca

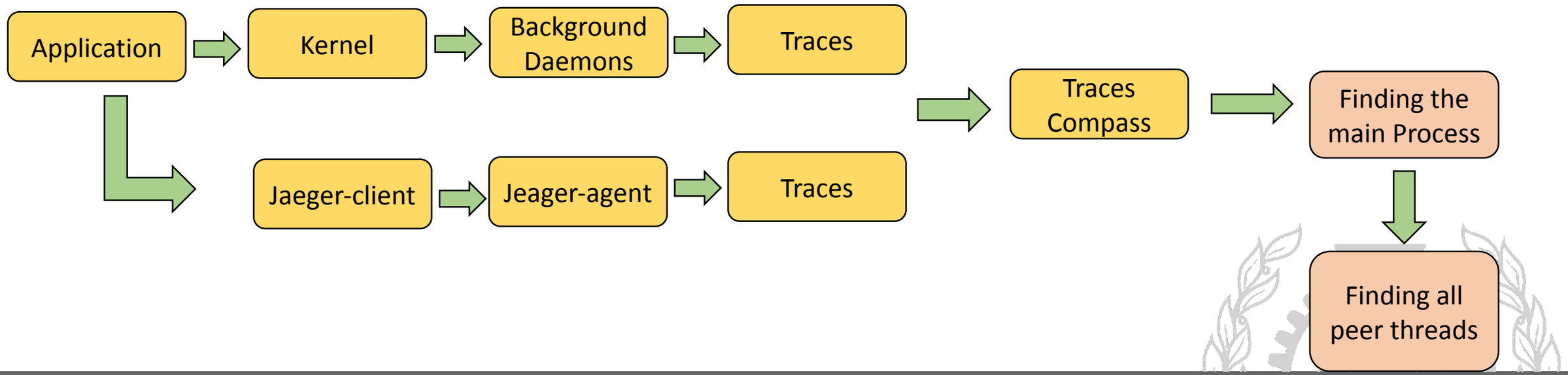
Using Performance Variation for Instrumentation Placement in Distributed Systems

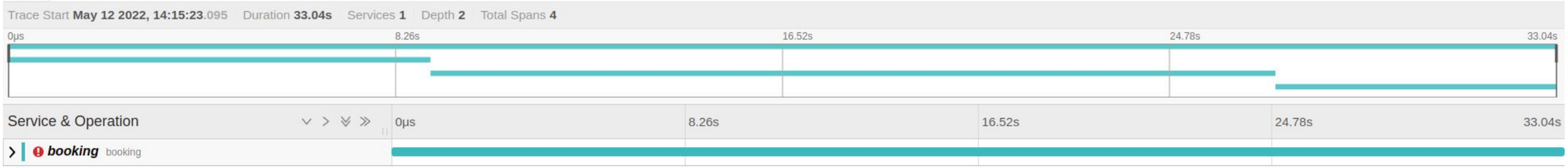
Roadmap

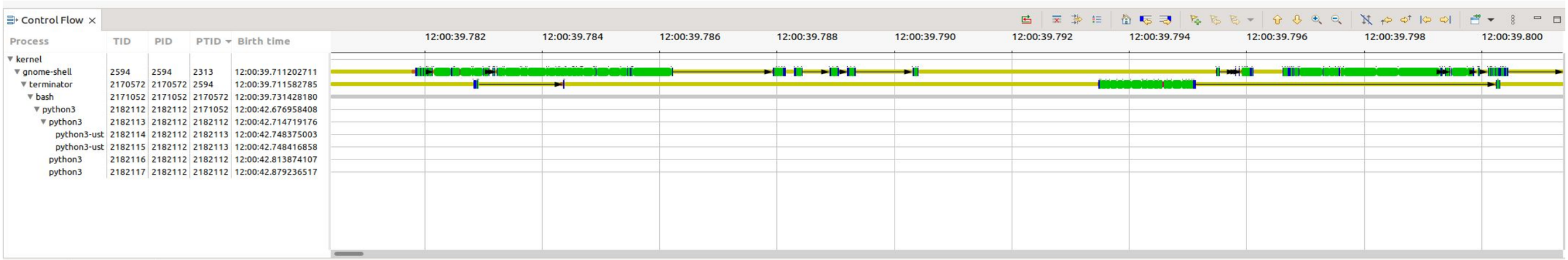

Trace Collection
Parsing Input
DAG Conversion
Critical Path Extraction
Clustering
Performance Comparision
Visualization

Trace Collection

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





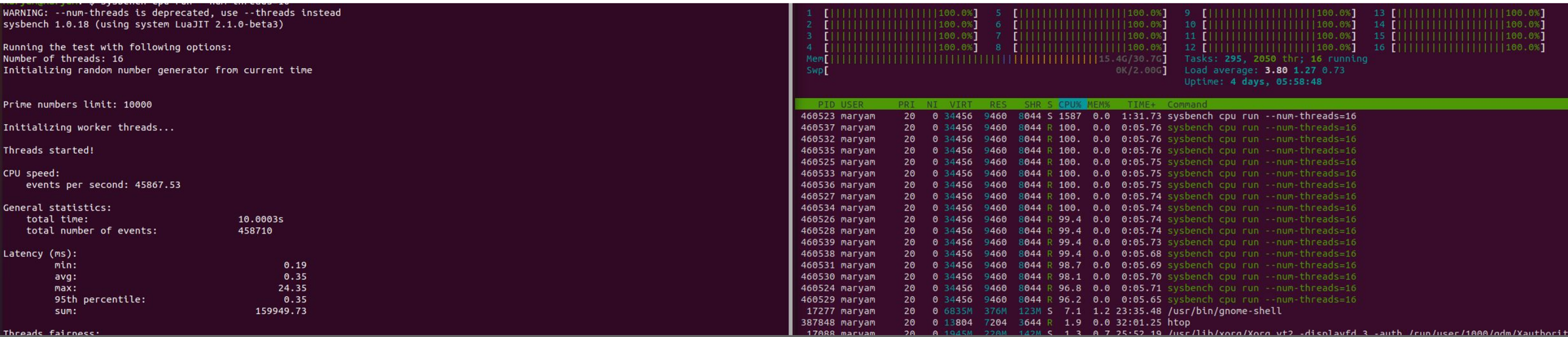


kernel | finishSpan | ust/uid/1000/64-bit

Trace	Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source	Binary Location	Fu
<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<s
kernel	12:00:39.704 534 967	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0cf6fb90000, bytes_req=739, bytes_alloc=1024, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 537 895	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0cf6fb90000, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 539 324	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce22dd0a00, bytes_req=312, bytes_alloc=512, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 540 225	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce22dd0a00, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 541 026	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce622d9d30, bytes_req=16, bytes_alloc=16, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 541 208	kernelchannel_0	0	power_cpu_idle	state=4294967295, cpu_id=0, context.packet_seq_num=0, context.cpu_id=0, context._tid=0, context._pid=0	0		0			
kernel	12:00:39.704 541 299	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce622d9d30, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 541 968	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 542 171	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 542 262	kernelchannel_0	0	power_cpu_idle	state=1, cpu_id=0, context.packet_seq_num=0, context.cpu_id=0, context._tid=0, context._pid=0	0		0			
kernel	12:00:39.704 542 525	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 542 697	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 542 849	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 001	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 153	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 316	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 468	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 619	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 771	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 543 934	kernelchannel_2	2	kmem_kfree	call_site=0xffffffffc14d9654, ptr=0xffffa0ce30e72c50, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			
kernel	12:00:39.704 544 086	kernelchannel_2	2	kmem_kmalloc	call_site=0xffffffffc14d95fe, ptr=0xffffa0ce30e72c50, bytes_req=2, bytes_alloc=8, gfp_flags=3264, context.packet_seq_num=0, context.cpu_id=2, context._tid=1050, context._pid=857	1050		857			

Strategy

- Local problem
  - Bad code
    - ✔ Infinite loop.
- System problem
  - System misconfiguration.
  - Workload on the CPU/ Memory/ Network.
    - ✔ Using benchmark tools for creating HW load.

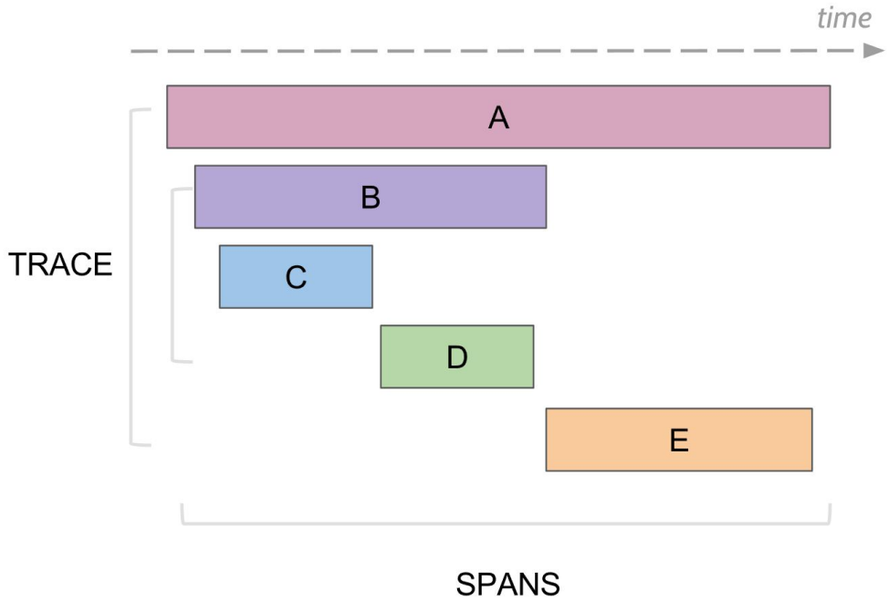


Strategy

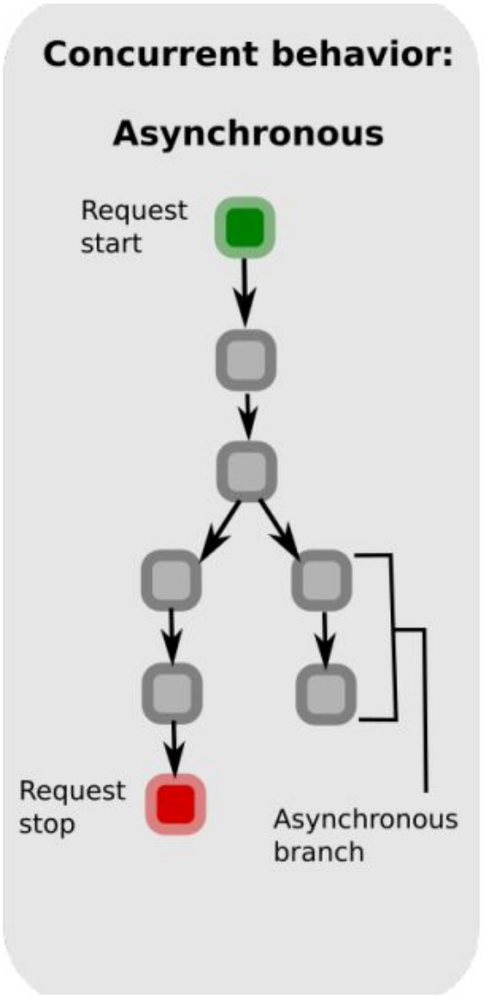
- Span or swim lane showing call graph relationship
- DAG also shows call graph

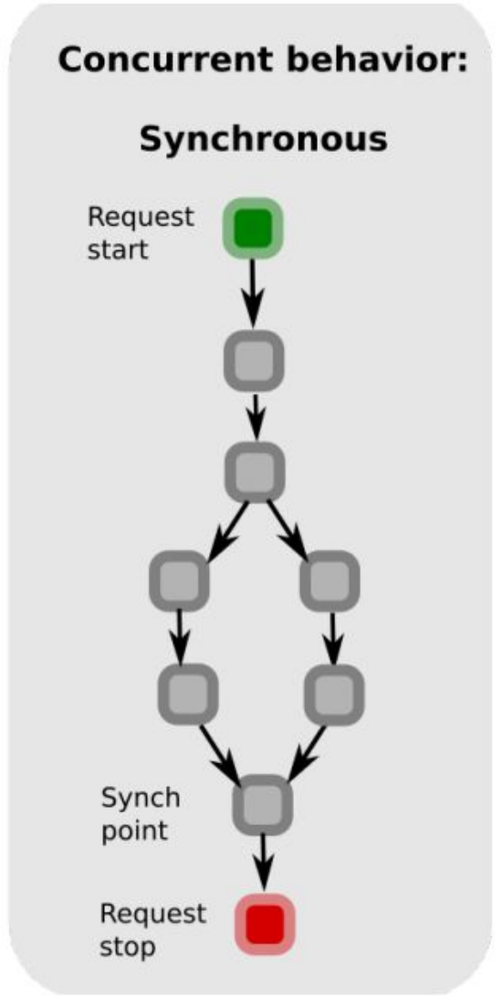
and the task's relationship

- Concurrent behavior
- Synchronization points
- Asynchronous Behavior



Reference:
https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37365088

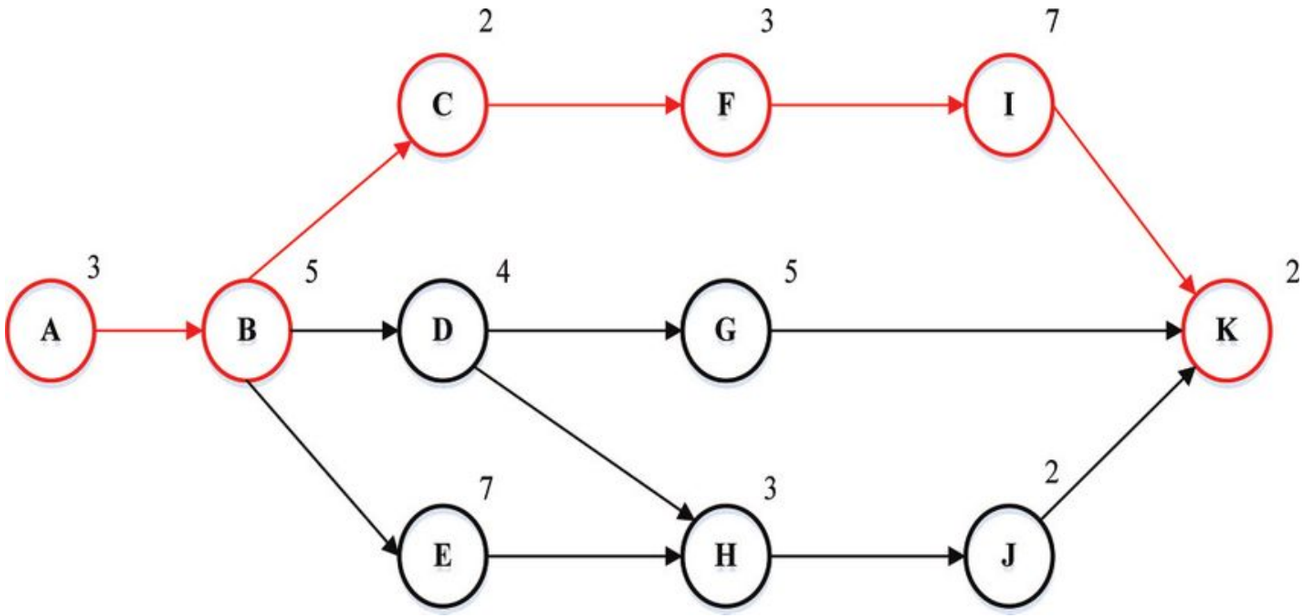




- Extracting critical path
  - Concurrency

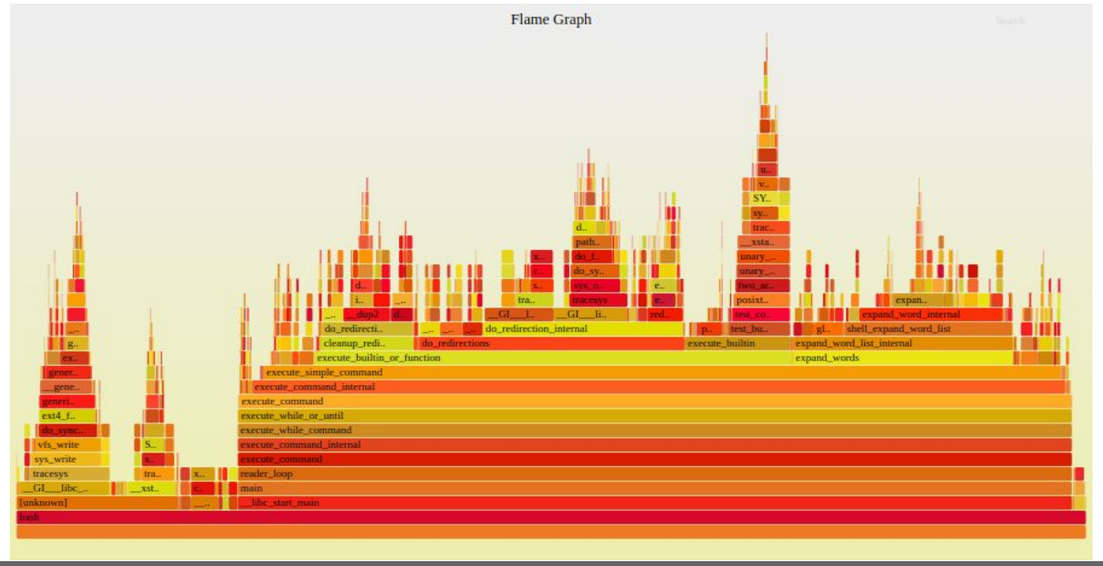
    Finding the path between the start and endpoint of the execution while we are not waiting for something else.
  - No concurrency

    The workflow of the application.



Strategy

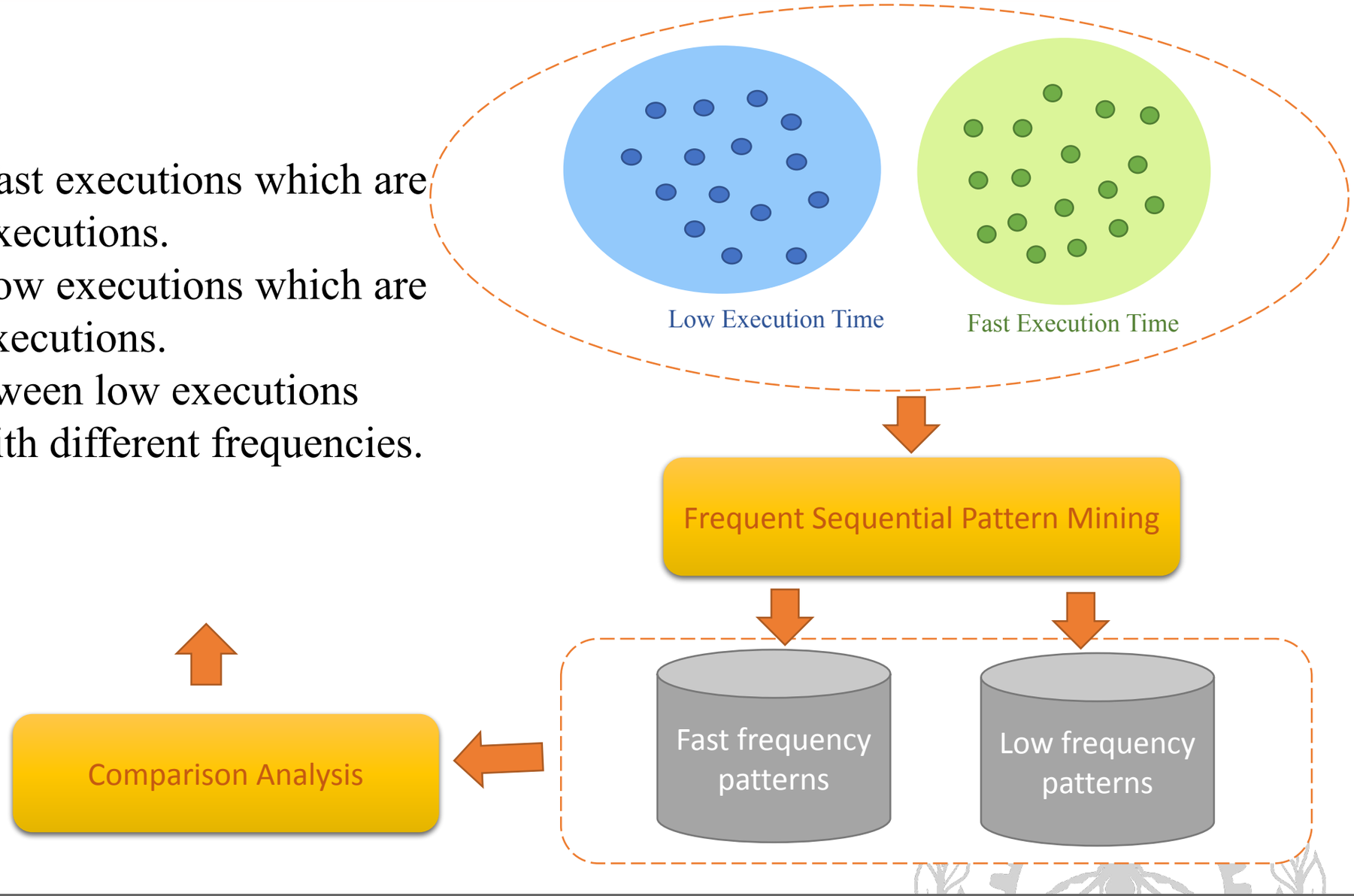
- Categorization
  - Thresholding
    - Number of page fault
    - Memory usage
    - Execution time
    - …
  - Clustering → Combination of performance metrics



Strategy

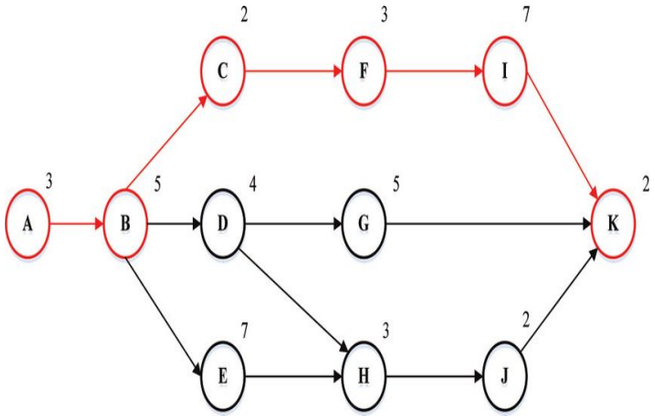
- Comparison:

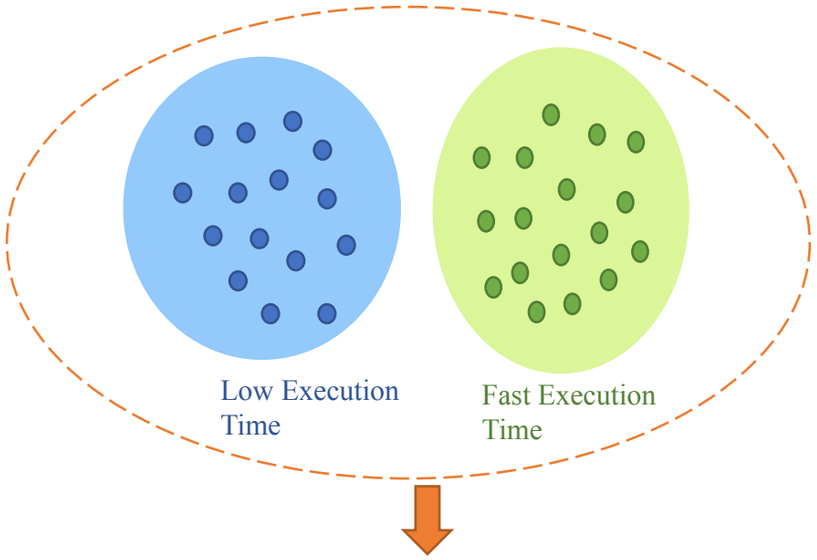
- Frequent patterns of fast executions which are not available in low executions.
- Frequent patterns of low executions which are not available in fast executions.
- Common patterns between low executions and fast executions with different frequencies.

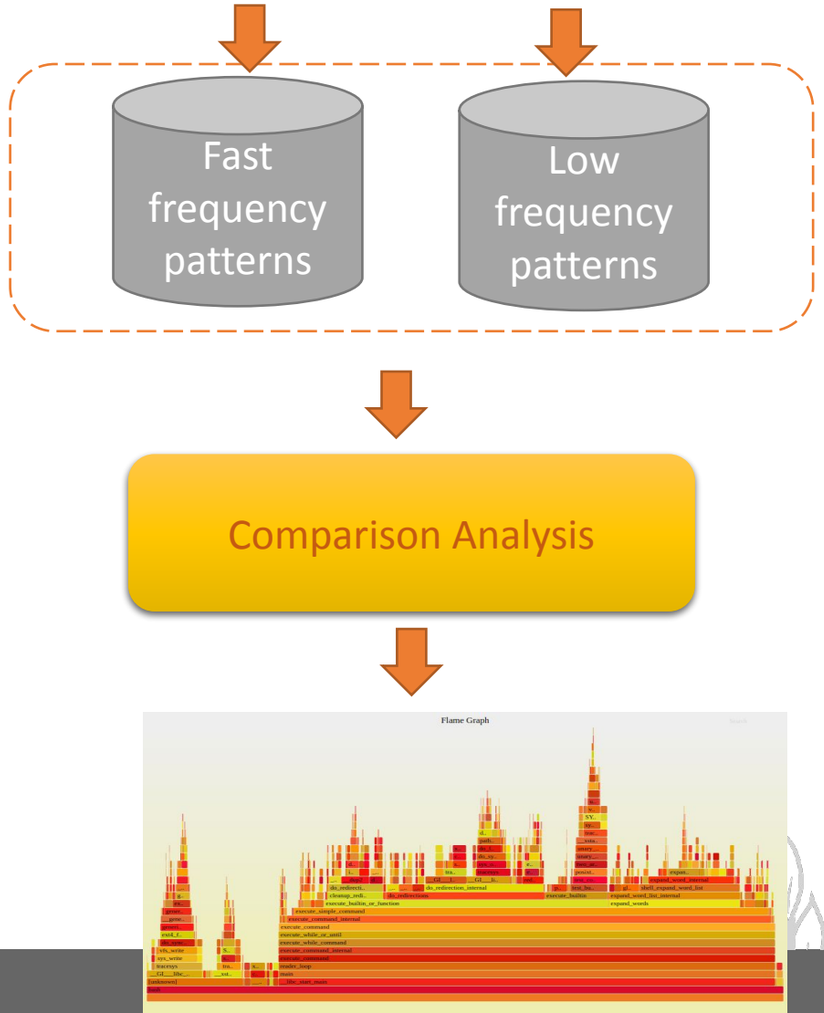


Future Goal

- Extracting critical path based on the DAG.
- Categorizing executions.
- Localizing performance problems.







Thank you

Email: maryam.ekhlasi@polymtl.ca

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