Performance Analysis of Data Stream Processing Systems like Apache Spark

POLYTECHNIQUE MONTRÉAL

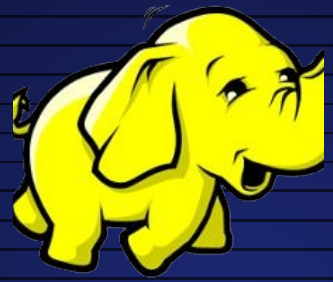
UNIVERSITÉ D'INGÉNIERIE

Reza Rouh

May 30th, 2024

Introduction/ Tools

Apache Spark



Apache Hadoop

Apache Flink

Apache Storm

Apache Kafka

Apache Beam

Introduction/ Apache tools logging system

	Spark	Hadoop	Flink	Storm	Kafka	Beam
Support log4j ?	✓	✓	✓	✓	✓	✓

They share similar logging mechanisms!

Introduction/Log4j

Type

Logging framework for Java logging ecosystem

Functionality

ERROR, WARN, INFO, DEBUG, TRACE

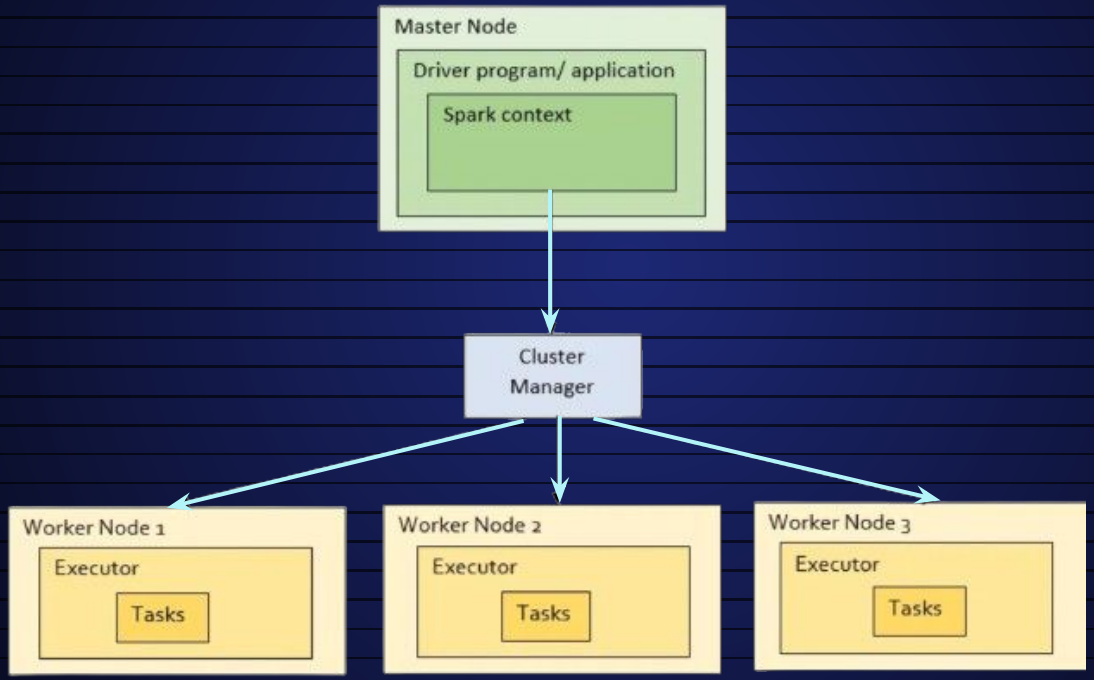
Customization

Allows detailed customization

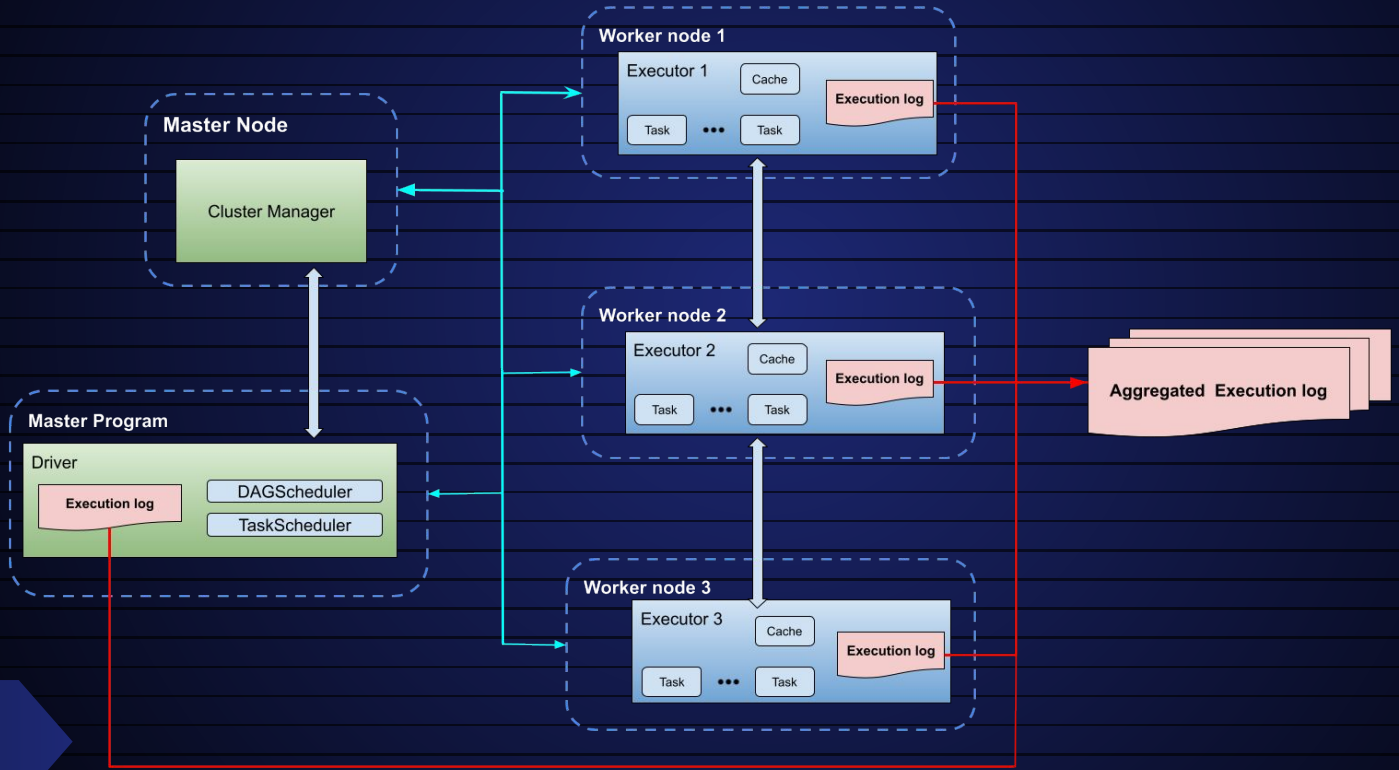
Versions

Log4j 1.x and Log4j 2.x

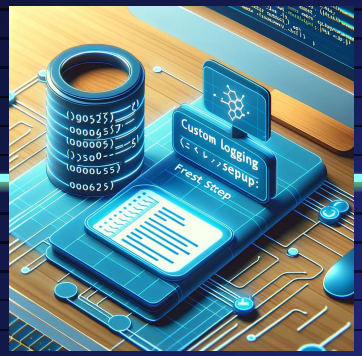
Introduction/ Spark Architecture



Logging mechanism



Objectives and Scope



Instrumentation

Set up

Logging

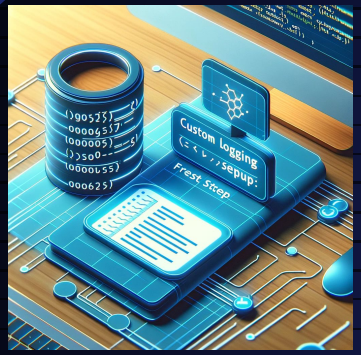
Analyze Traces

Methodology

- Manage default Log4j logs
- Adding new instrumentation

Instrumentation

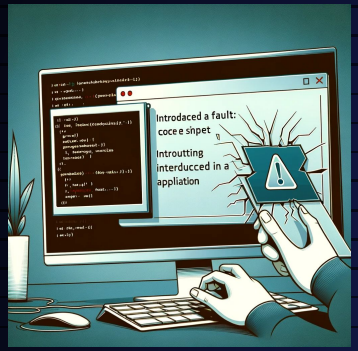
Methodology



- Setting up Apache Spark for new logs
- Setting up Apache Spark for LTTNG
- Setting up LTTNG in Spark cluster on Docker

Set up

Methodology



- Capture UST events by LTTNG
- Capture kernel events by LTTNG

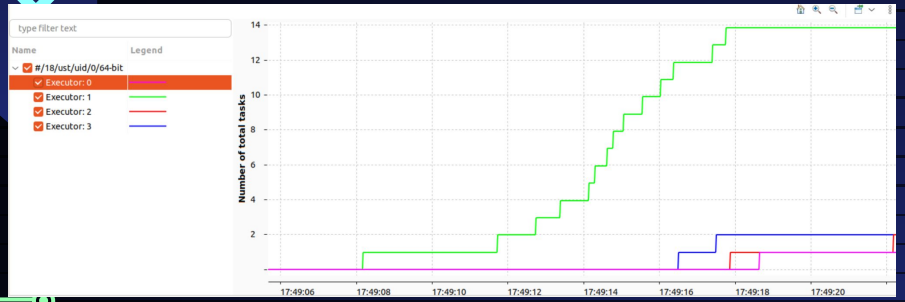
Logging

Methodology

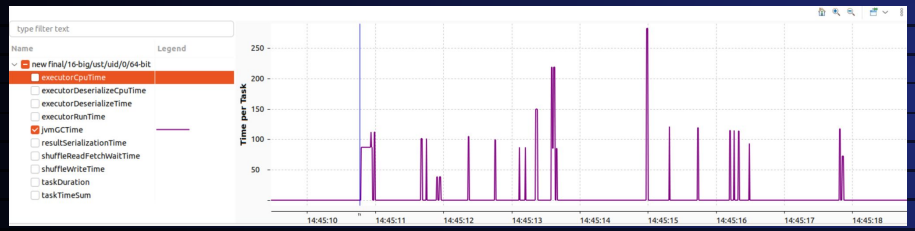
- Clean up traces
- Analyze traces
- Analyze and Visualize in Trace Compass

Analyze Traces

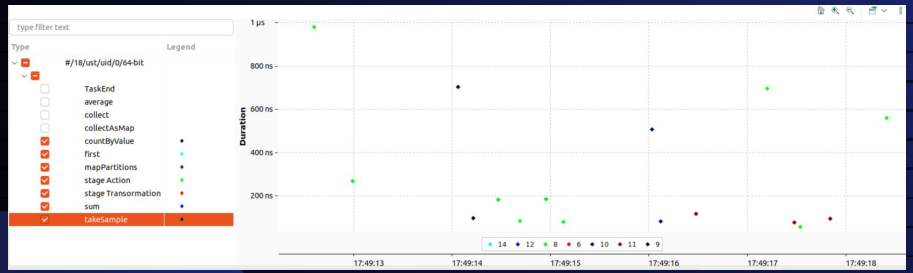
UST Visualization



Number of task per Executors

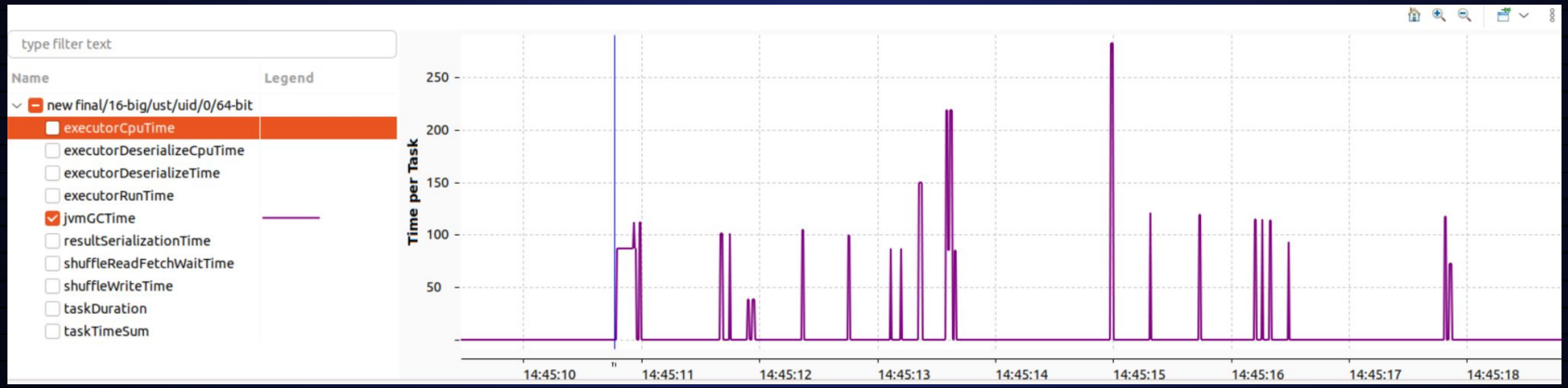


Times parameters per Task



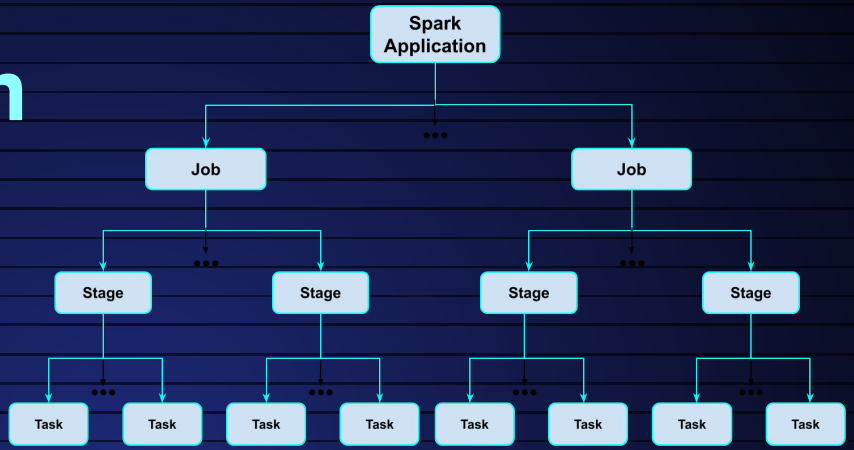
Time Duration for Task Operations

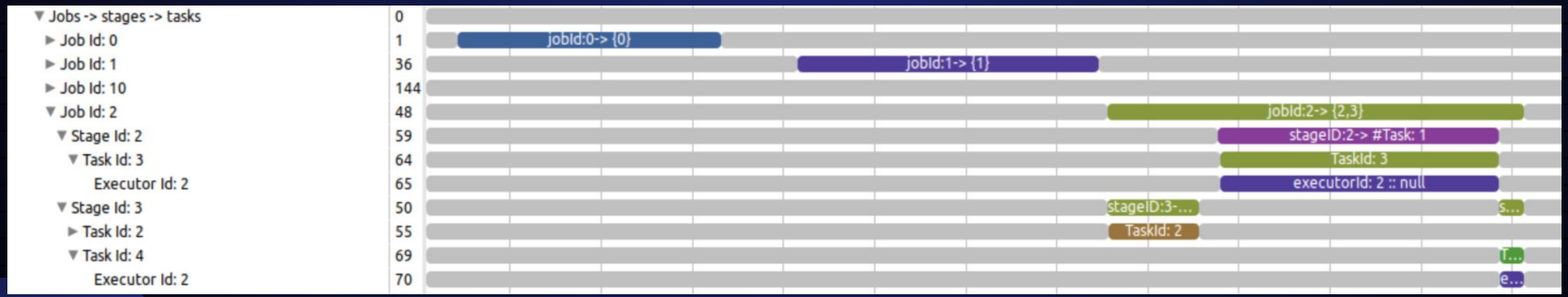
UST Visualization



Times parameters per Task XY graph

UST Visualization

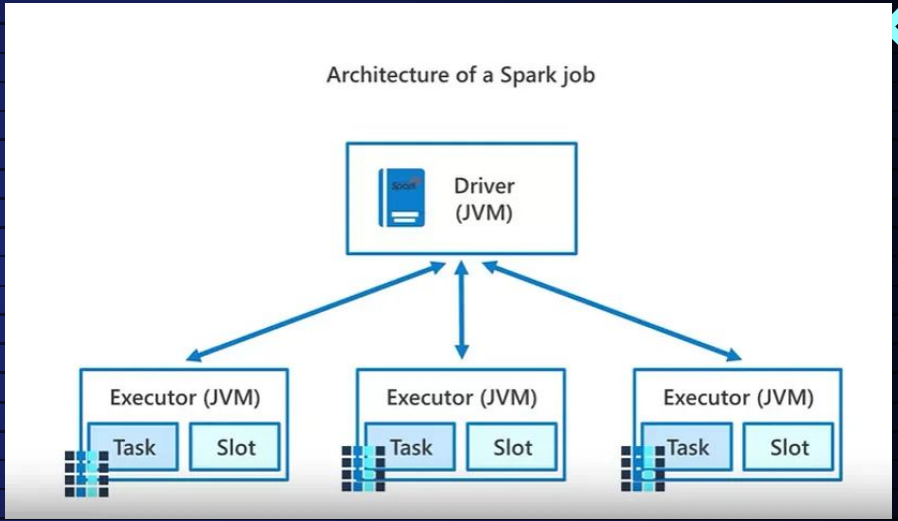




Job stage task hierarchy

A big CHALLENGE for kernel analysis

1. Executors run on JVM
2. We need kernel events for each Task
3. How to correlate these?



CPU	Event Type	TID	Contents
3	lttng_log4j:event	253	-
1	sched_waking	50123	-
1	sched_switch	50123	-
5	sched_switch	50122	-
3	timer_hrtimer_start	50121	-

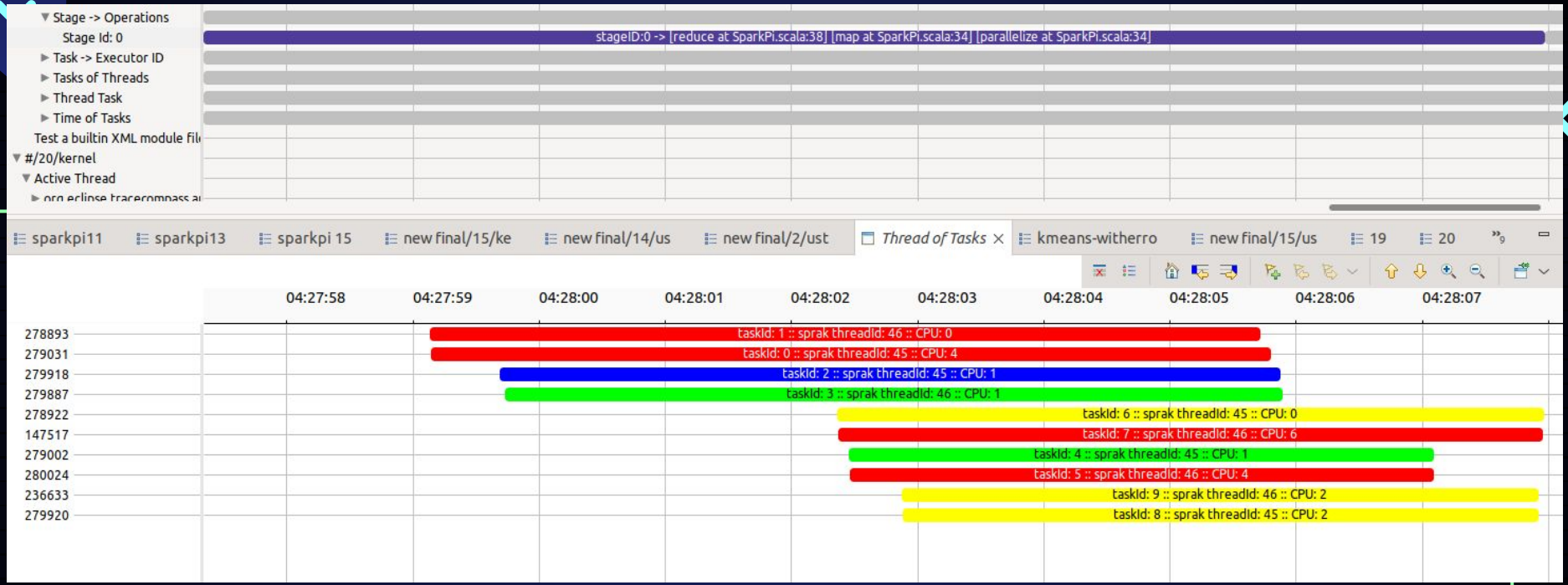
instrumentation

3	sched_switch	50121	prev_tid=50121

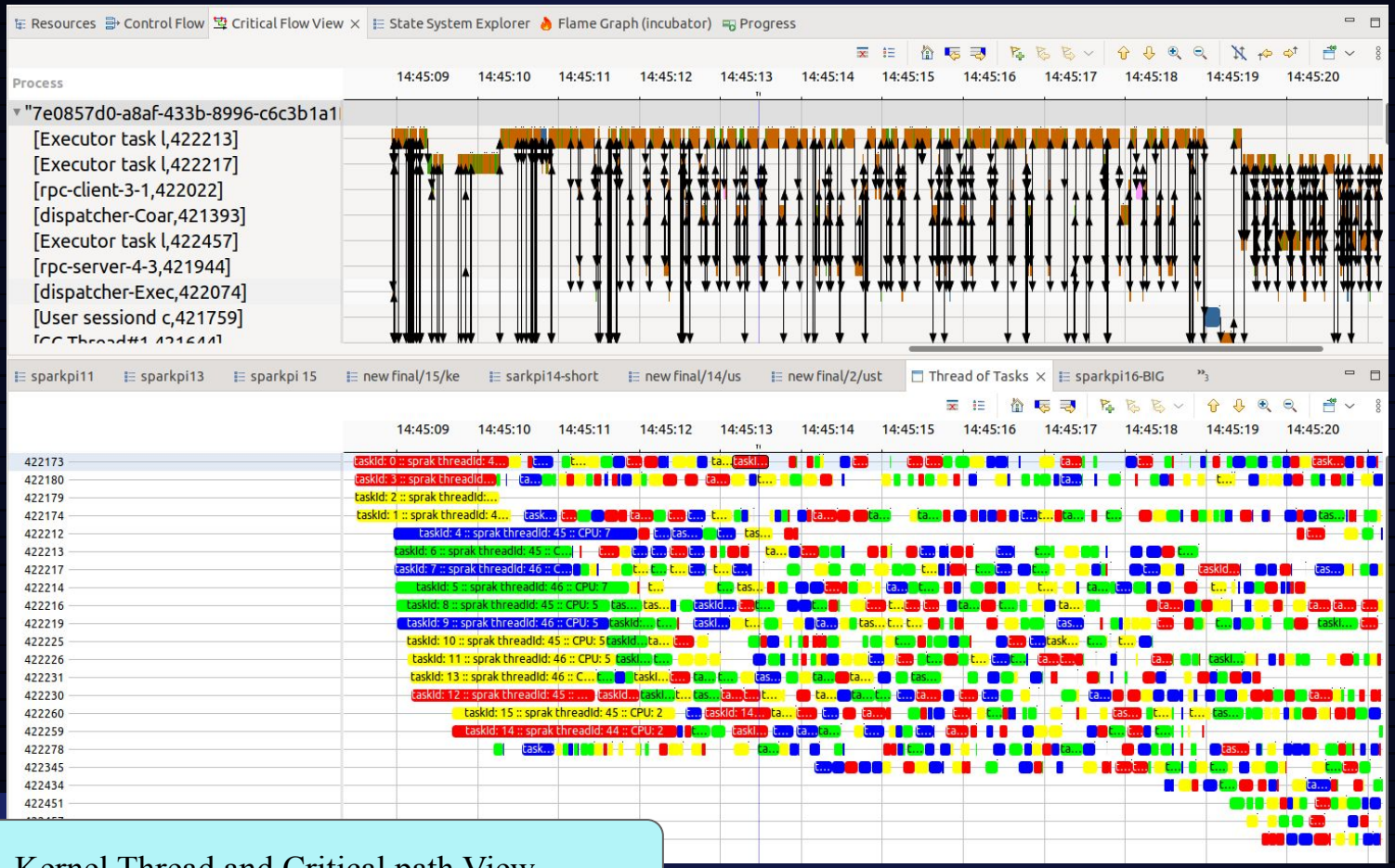
Get thread ID from first kernel event after Spark task start event from **same CPU**

CPU	Event Type	TID	Contents
3	lttng_log4j:event	253	-
1	sched_waking	50123	-
1	sched_switch	50123	-
5	sched_switch	50122	-
3	timer_hrtimer_start	50121	-
3	sched_switch	50121	prev_tid=50121

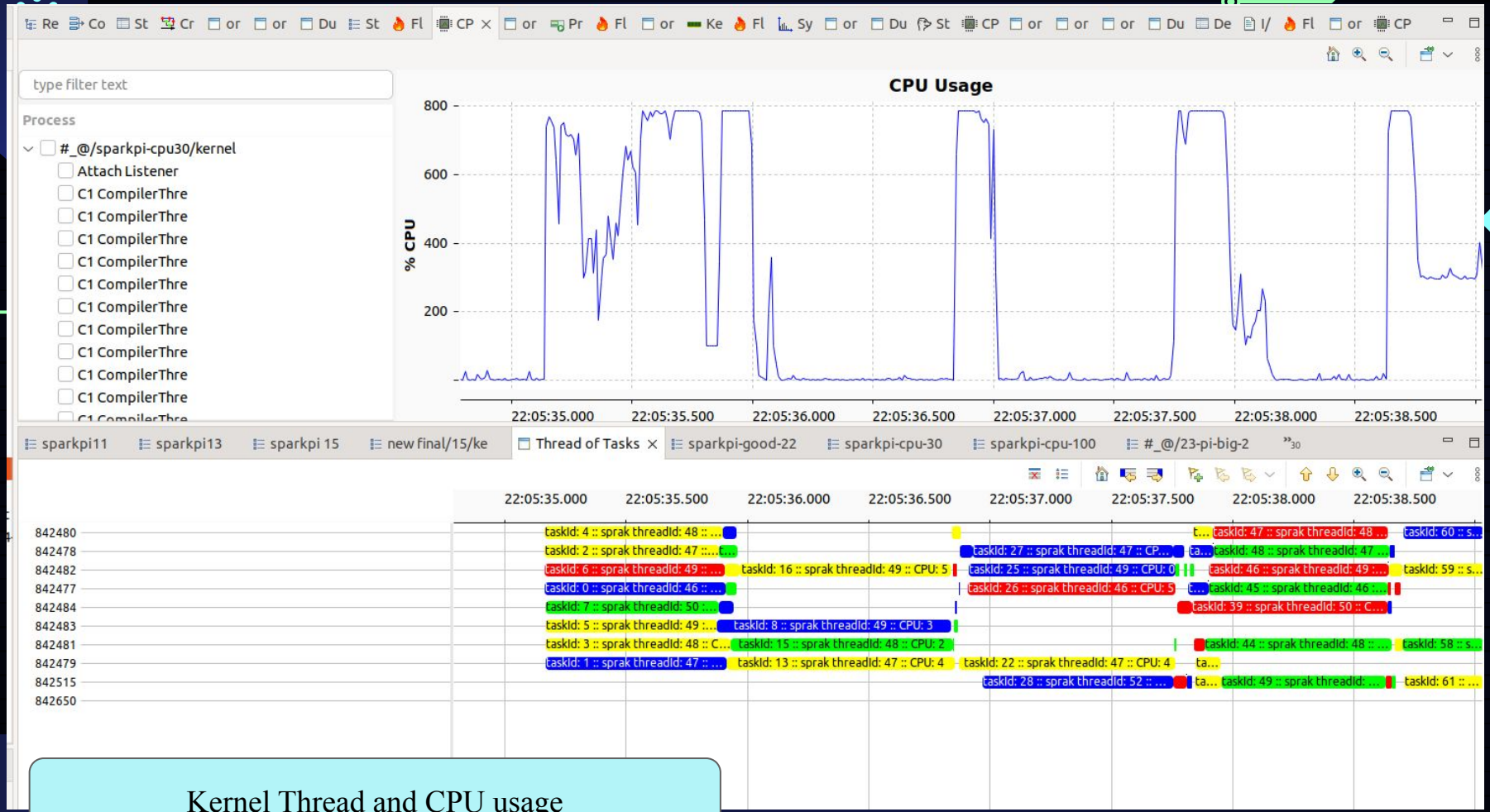
Get thread ID from first sched_switch after Spark task start event from **same CPU**



Kernel Thread and Task Operations correlation

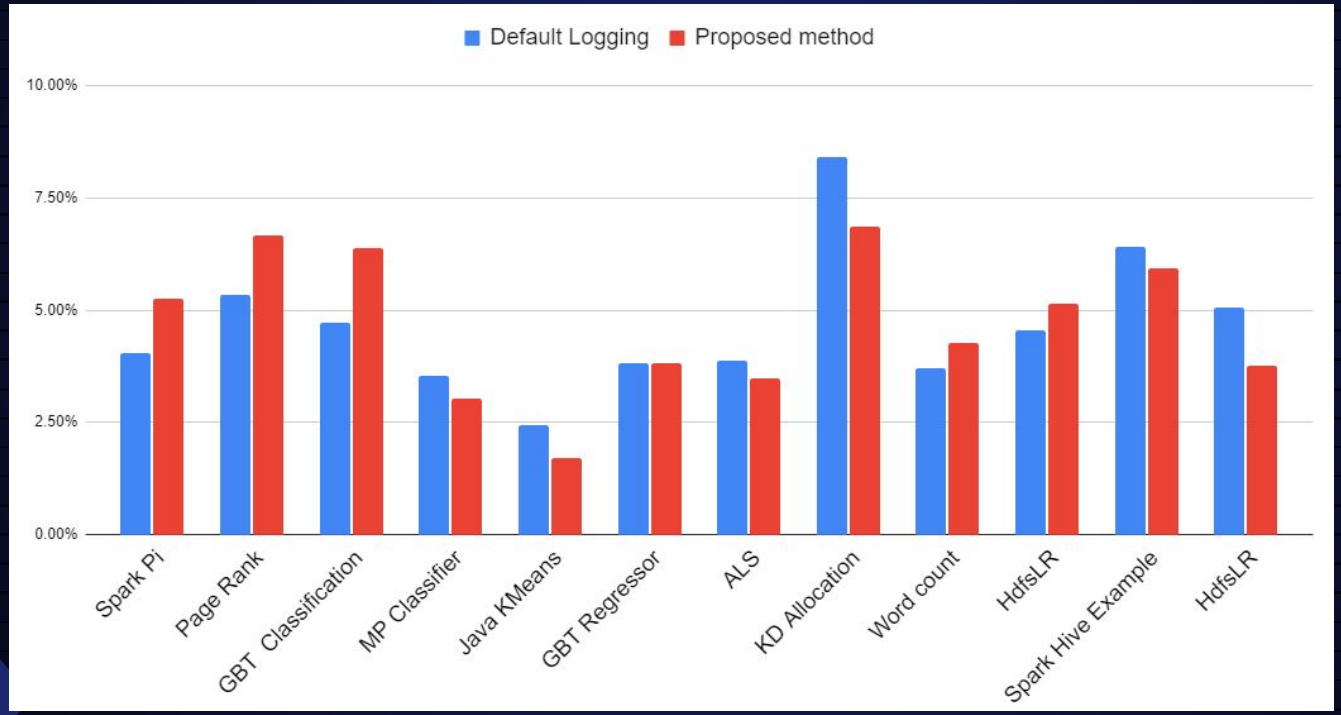


Kernel Thread and Critical path View



Kernel Thread and CPU usage

Execution Time overhead



Future Work

1. Using a new tracer
2. Automatic detection
3. Have more analysis because now we have kernel events!


THANK YOU!
Any questions?