PERFORMANCE ANALYSIS OF CONTAINER-BASED ARCHITECTURES

Pierre-Frédérick DENYS

Friday 15 May 2020

Agenda

- Introduction
- Messaging monitoring in a container-based environnement
- Trace viewers for containers architecture
- Conclusion

Introduction

- No specific tools for performance analysis of container-based architecture

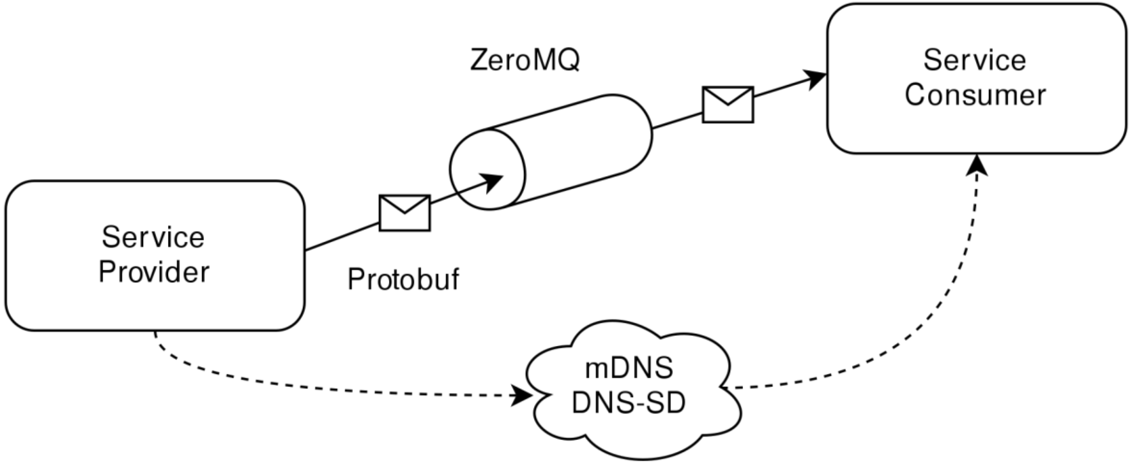
- Lack of flexibility in trace viewers to make efficient analysis for thses architectures

Part 1 : containers messaging analysis

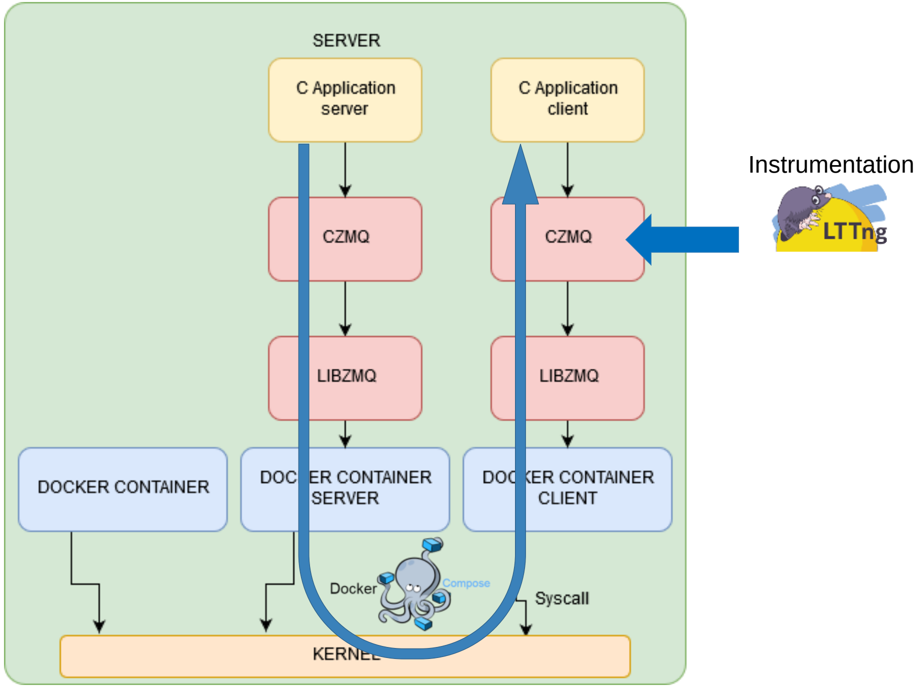
“ How to monitor messaging between containers and prevent data loss ?

CASE STUDY : zeroMQ as message broker

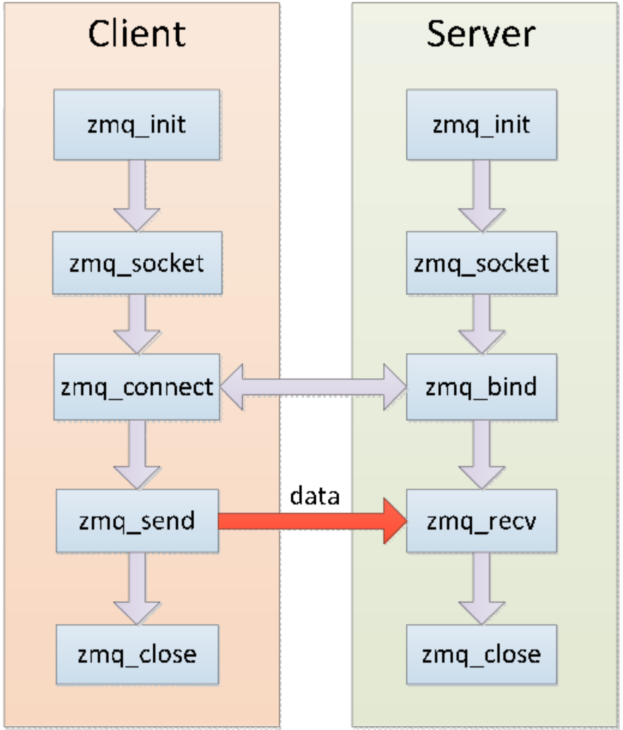
ØMQ



CASE STUDY : zeroMQ as message broker



CASE STUDY : zeroMQ as message broker



CASE STUDY : zeroMQ as message broker

What is done actually :

- Data collection at kernel level, libzmq level, czmq level (actual solution)
- Patch for ZeroMQ (ZeroMQ instrumentation and LTTng data collection)

What I will work on :

- Test this patch in a real architecture
- Implement a mechanism to activate and deactivate the instrumentation without recompiling the library
- Get more data with the tracepoints (on the sender, on the message destination)
- Generalization to other asynchronous messaging systems.

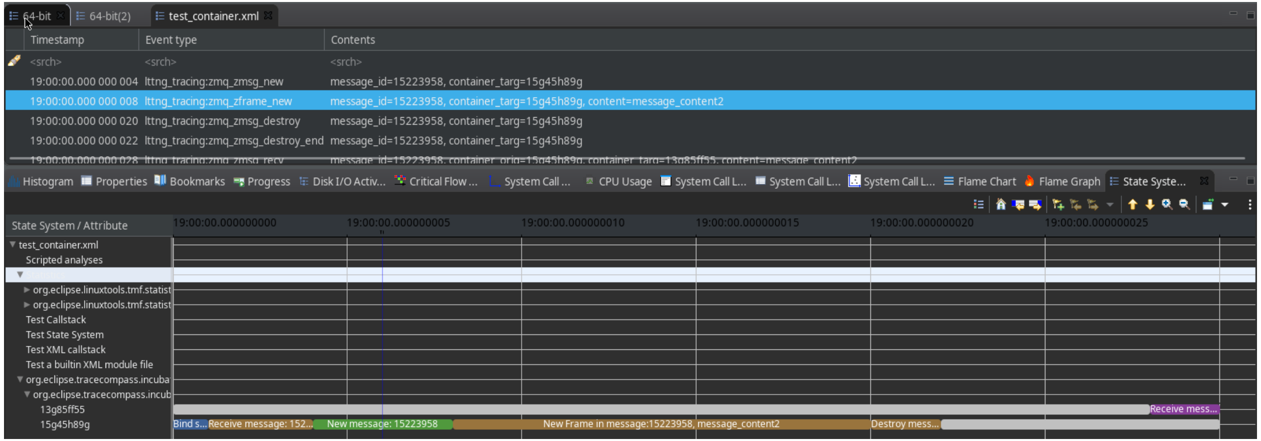
“ Part 2 : Trace viewers for containers architectures

“ How to display the result of tracing sessions on container messaging ?

Trace Compass as a viewer

- Trace compass is used to display traces for the moment :

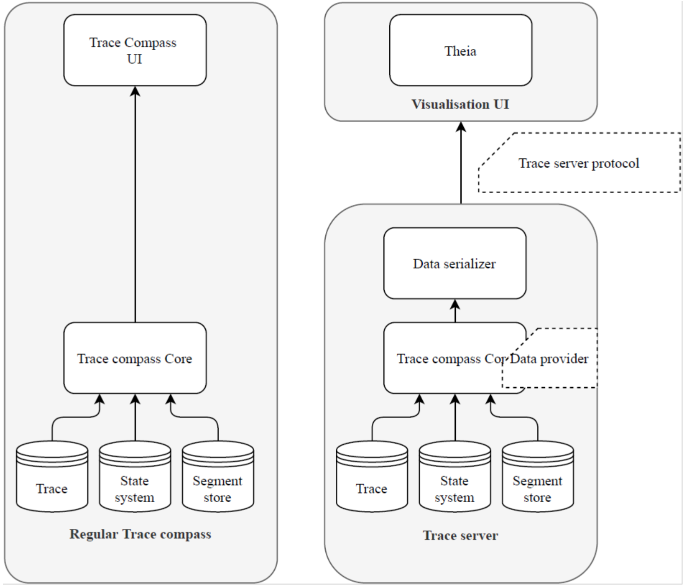
ZMQ events associated with the message (new message, new frame, send on sockets) on the sender and on the receiver.



Timestamp	Event type	Contents
<srch>	<srch>	<srch>
19:00:00.000 000 004	lttng_tracing:zmq_zmsg_new	message_id=15223958, container_targ=15g45h89g
19:00:00.000 000 008	lttng_tracing:zmq_zframe_new	message_id=15223958, container_targ=15g45h89g, content=message_content2
19:00:00.000 000 020	lttng_tracing:zmq_zmsg_destroy	message_id=15223958, container_targ=15g45h89g
19:00:00.000 000 022	lttng_tracing:zmq_zmsg_destroy_end	message_id=15223958, container_targ=15g45h89g
19:00:00.000 000 028	lttng_tracing:zmq_zmsg_recv	message_id=15223958, container_orig=15g45h89g, container_targ=13g85ff55, content=message_content2

- Lack of flexibility in the views
- Difficulty of customization by the end user without re-compiling Trace Compass

Possibilities with the trace server of Theia



- With the new architecture of Trace Compass implemented for Theia :

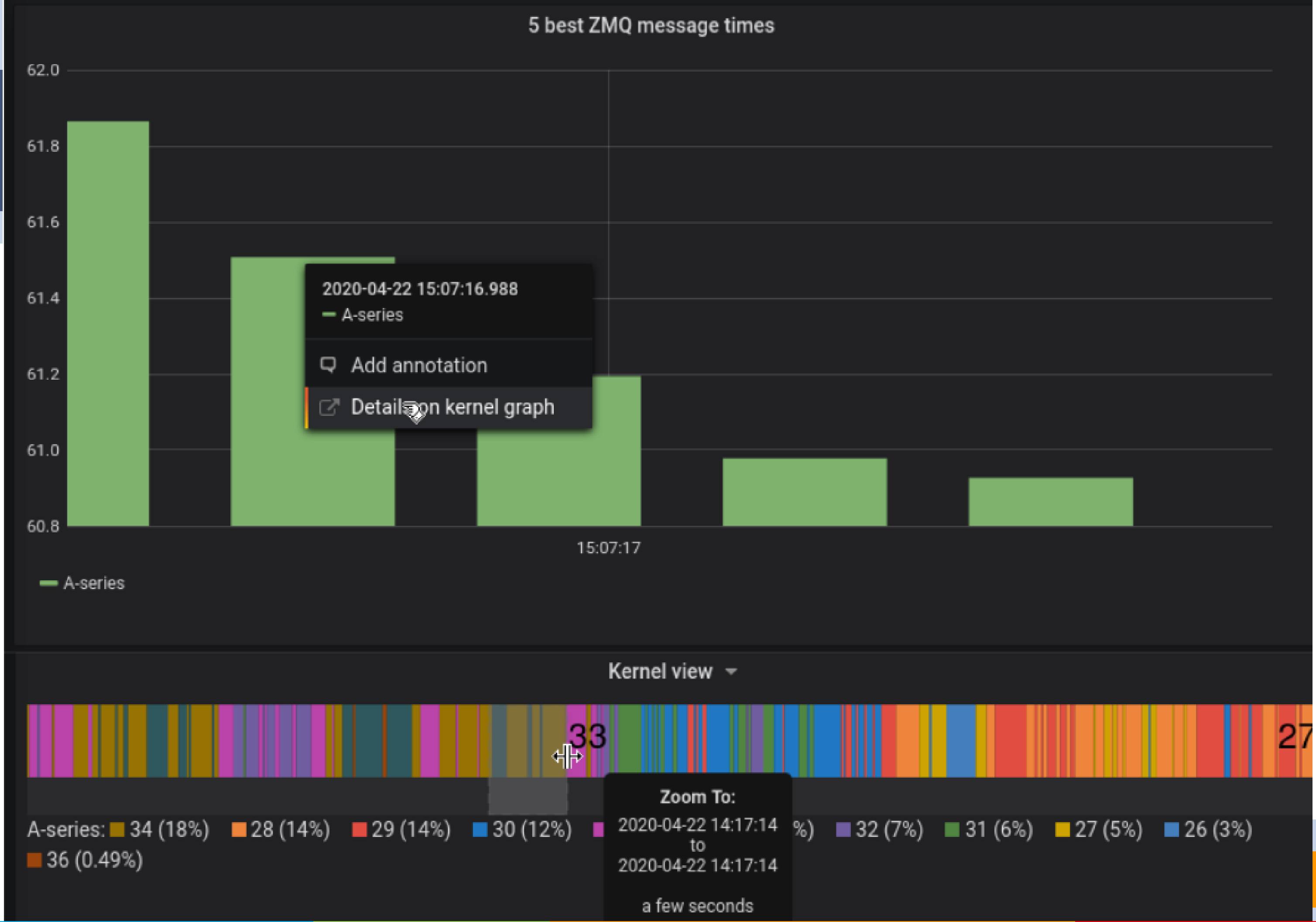
- A lot of possibilities with the REST API of trace server (Trace server Protocol)
- Use the Trace server as a backend for other data viewers

Let’s start with Grafana

Grafana

- Grafana is an open source data viewer widely use in companies
- Online solution
- Easy to use and lots of customization in the web interface
- Dynamic graphs, with links and redirections
- Grafana is designed for metrics analysis

- How to offer an alternative to Theia ?
- Need a module to convert TSP to a JSON compatible format
- Need to implement a type of graph for kernel view


5 best ZMQ message times
62.0
61.8
61.6
61.4
61.2
61.0
60.8
15:07:17
A-series
2020-04-22 15:07:16.988
A-series
Add annotation
Details on kernel graph
Kernel view
33
27
Zoom To:
2020-04-22 14:17:14
to
2020-04-22 14:17:14
a few seconds
A-series: 34 (18%)
28 (14%)
29 (14%)
30 (12%)
%)
32 (7%)
31 (6%)
27 (5%)
26 (3%)
36 (0.49%)

Trace viewers

In progress :

- Module to convert TSP to Grafana compatible JSON

Work to do :

- Define a standard for compatible JSON (event links…)
- Kernel view module for Grafana

- Add similar views to Theia

Conclusion

- Tools to monitor messaging in a container-based environnement
  - ZeroMQ instrumentation
- Improve tools for container messaging analysis
  - Offer an alternative to Theia

Thank you for listening !