End to End tracing in Ceph

Project Developers : Golsana Ghaemi, Oindrilla Chatterjee, Aditya Singh, Bowen Song
Mentors : Mania Abdi, Raja Sambasivan, Peter Portante

Project Overview

Ceph - Existing Tracing - Blkin

Distributed Storage System

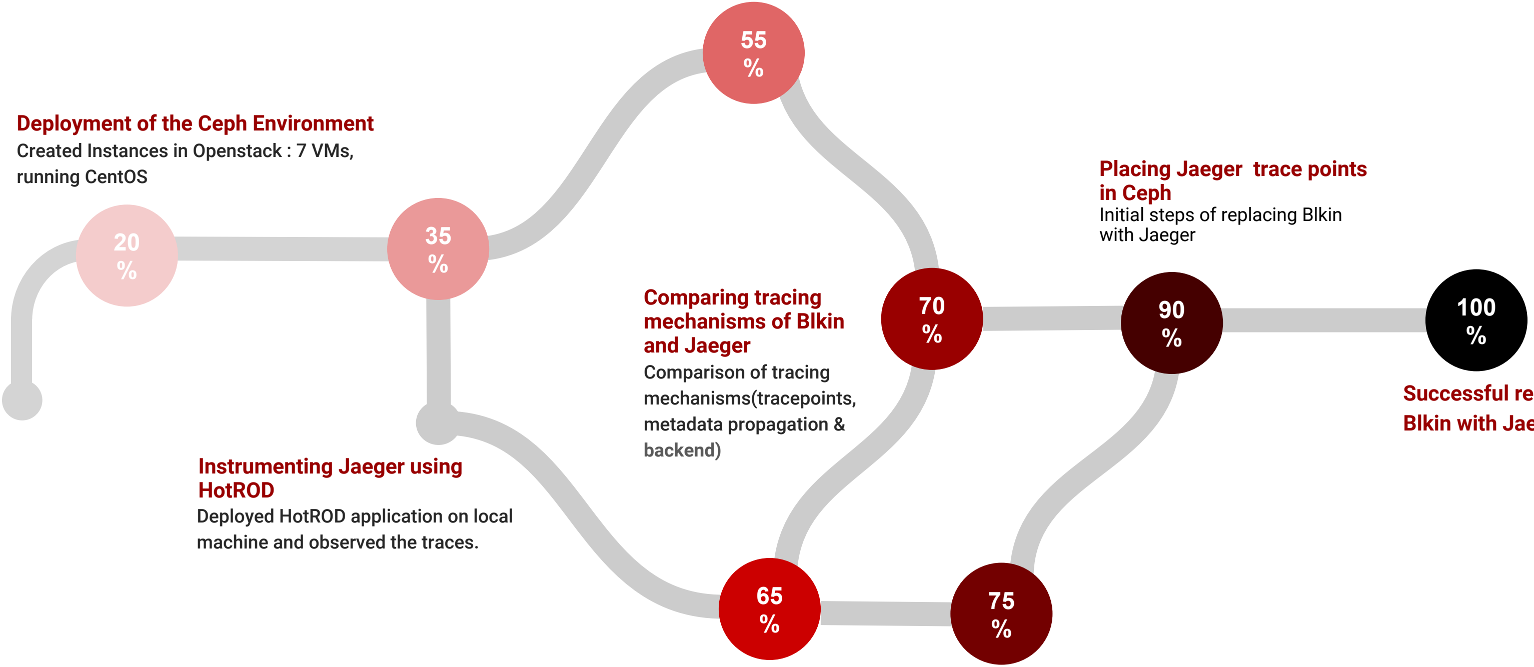
Uber - Tracing - Jaeger

Distributed Tracing System - Always on, End-to-End

Ceph - Tracing - Jaeger

Distributed Storage System + Distributed Tracing System

Key Actions



Deployment of the Ceph Environment
Created Instances in Openstack : 7 VMs, running CentOS

Instrumenting Jaeger using HotROD
Deployed HotROD application on local machine and observed the traces.

Compiling Ceph with Blkin
On a virtual cluster with 3 OSDs, 1 Mon node, 1 Metadata

Exploring HOTROD example
Deployed HotROD on two VMs in MOC in order to visualize the traces using Jaeger

Comparing tracing mechanisms of Blkin and Jaeger
Comparison of tracing mechanisms(tracepoints, metadata propagation & backend)

Deploy Jaeger on a Distributed Client Server Application
Instrumenting Jaeger on a simple testbed application similar to Ceph

Placing Jaeger trace points in Ceph
Initial steps of replacing Blkin with Jaeger

Successful replacement of Blkin with Jaeger.

LTTng

LTTng (*Linux Trace Toolkit*) is an open source software toolkit which can be used to simultaneously trace the Linux kernel, user applications, and user libraries.

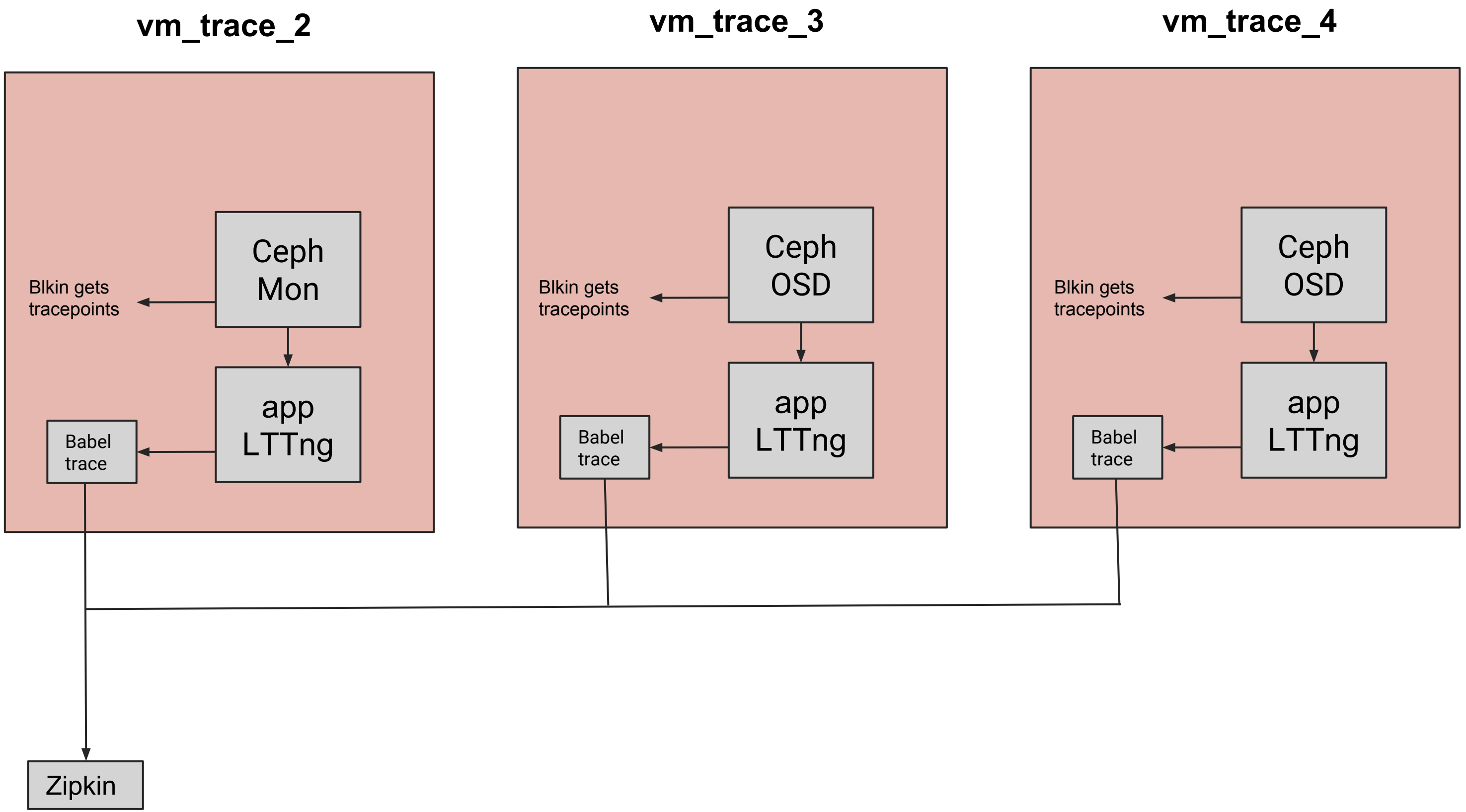
LTTng has four primary tracing domains

- Linux kernel
- User space
- java.util.logging (JUL)
- Python

It can do tracing in these domains. The tracepoints based on the LTTng format are added to the desired applications. Its libraries are included in applications.

Blkin works based on LTTng and uses its libraries. Actually, Blkin gets tracepoints and sends them to LTTng.
Blkin saves traces in files. Babeltrace is used to convert LTTng traces to Zipkin and merge them.
Zipkin is used to visualize the trace.

LTTng


vm_trace_2
vm_trace_3
vm_trace_4
Ceph Mon
Ceph OSD
Ceph OSD
Blkin gets tracepoints
Blkin gets tracepoints
Blkin gets tracepoints
app LTTng
app LTTng
app LTTng
Babel trace
Babel trace
Babel trace
Zipkin

Using Blkin to trace Ceph

- Compiling and running Blkin
- Compiling and running Ceph
- Using virtual cluster with 3 OSDs, one monitor node and one metadata as a test environment to capture ceph traces

Problems we encountered:

- Version of Ceph
- Compiling Blkin with Ceph
- Errors in Ceph after compiling Blkin
- Difficulty in tracing with Blkin : Start and Stop

Deployment of HotROD and Jaeger

Current Status:

- Environment Setup - GO, Docker
- Display Setup - GUI Interaction: Firefox, Text based web browser
- Authentication Setup - sshd, X11
- Learning Example - HotROD, Distributed GO language chat service

Next Plan:

- Setup Distributed GO language chat service on 2 VMs
- Put trace points in Distributed GO language chat service
- Language Wrapper, GO g++ Compile friendly
- Put trace points in Centos

Meaning of Traces in Jaeger

```
2018-03-14T18:53:57.346-0400	INFO	log/spanlogger.go:34	Found customer	{"service": "frontend", "customer": {"ID":"392","Name":"Trom Chocolatier","Location":"577,322"}}

2018-03-14T18:53:57.712-0400	INFO	log/spanlogger.go:34	Finding route	{"service": "frontend", "component": "route_client", "pickup": "368,94", "dropoff": "577,322"}

2018-03-14T18:53:57.404-0400	ERROR	log/spanlogger.go:39	redis timeout	{"service": "driver", "driver_id": "T742136C", "error": "redis timeout"}
github.com/jaegertracing/jaeger/examples/hotrod/pkg/log.spanLogger.Error
	/Users/bowensong/go/src/github.com/jaegertracing/jaeger/examples/hotrod/pkg/log/spanlogger.go:39
…
/Users/bowensong/go/src/github.com/jaegertracing/jaeger/vendor/github.com/uber/tchannel-go/inbound.go:195
```

Blkin trace of Ceph

```
[11:31:22.404471520] (+0.000039067) vm-trace-1.moclocal zipkin:timestamp: { cpu_id = 0 }, {
trace_name = "osd op", service_name = "osd.1", port_no = 0, ip = "0.0.0.0", trace_id =
1372444998004577892, span_id = 7588510525472714565, parent_span_id =
3824851228694617424, event = "enqueue op" }
```

- Time of VMs
- Trace_name: we are tracing osd : osd op
- Each command (get, put, …) is list of events and consists of multiple requests
- Captured event in osd op: event = "enqueue op"
- Trace_id per request : hard to propagate
- Each command consists of multiple components : service_name
- Going through each component: span, one span_id
- With parent_span_id, span_id, and trace-id we can make span tree

HotROD and Jaeger Challenges encountered

- SSH authentication error - Prevents us from entering the VMs setup on MOC

- Apps on VMs visualization - GUI needed to visualize the traces. We used X11 forwarding to forward the display of the remote application.

- Understanding of App deployment on server (ROOT directory)

- Distributed Software deployment - The two individual deployments on the VMs need too be linked to each other.

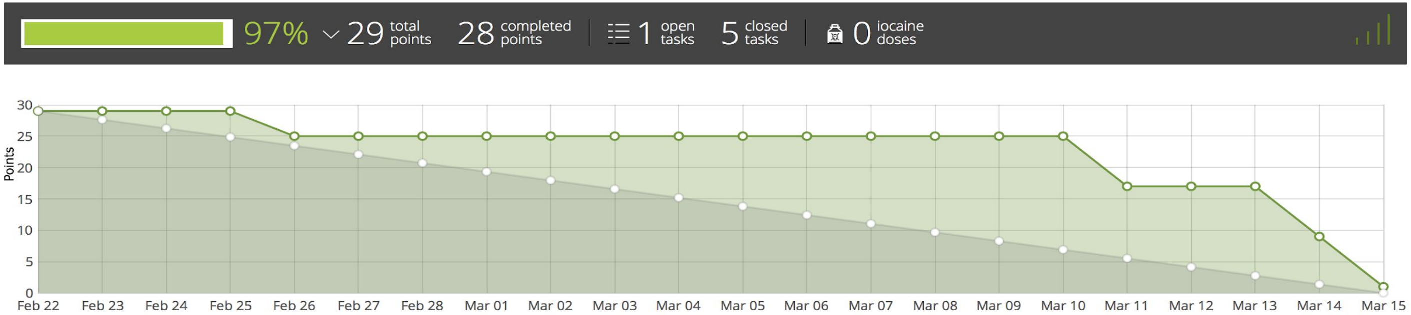
User Stories

Sprint 3(Current)

- As a product owner I should set the Radosgw client so that we can demonstrate the metrics of the Ceph deployment and its interaction with the clients
- As a product owner, I should learn the Blkin architecture and how it is going about the tracing in Ceph currently and its procedure for implementing the three parts of tracing
- As a product owner, I should compile a project presentation, so that the audience and stakeholders have a good understanding of the progress and workflow of our team
- As a product owner, I would like to compile Ceph with Blkin so that I understand the existing limitation

Sprint 4(Next)

- As a product developer, I would like to Deploy a GO language distributed application on 2 VMs, add trace points, and visualize the traces so that I have an understanding of instrumenting jaeger on a simple distributed application similar to Ceph
- As a product owner, i would like to Understand the architecture of Blkin to identify the tracing mechanism (tracepoints, metadata propagation, backend) and compare with Jaeger
- As a product developer, I would like to carry out the Initial steps in replacement of Blkin with jaeger so that I take my first important step towards my end goal



References

[1] https://lttng.org/docs/v2.10/#doc-whats-new

[2] Red Hat, Inc. (2017) Ceph homepage. [Online]. Available: https://ceph.com

[3] Red Hat, Inc. (2016) Tracing Ceph With BlkKin Ceph Documentation. [Online]. Available: http://docs.ceph.com/docs/master/dev/blkin/

Thank You!