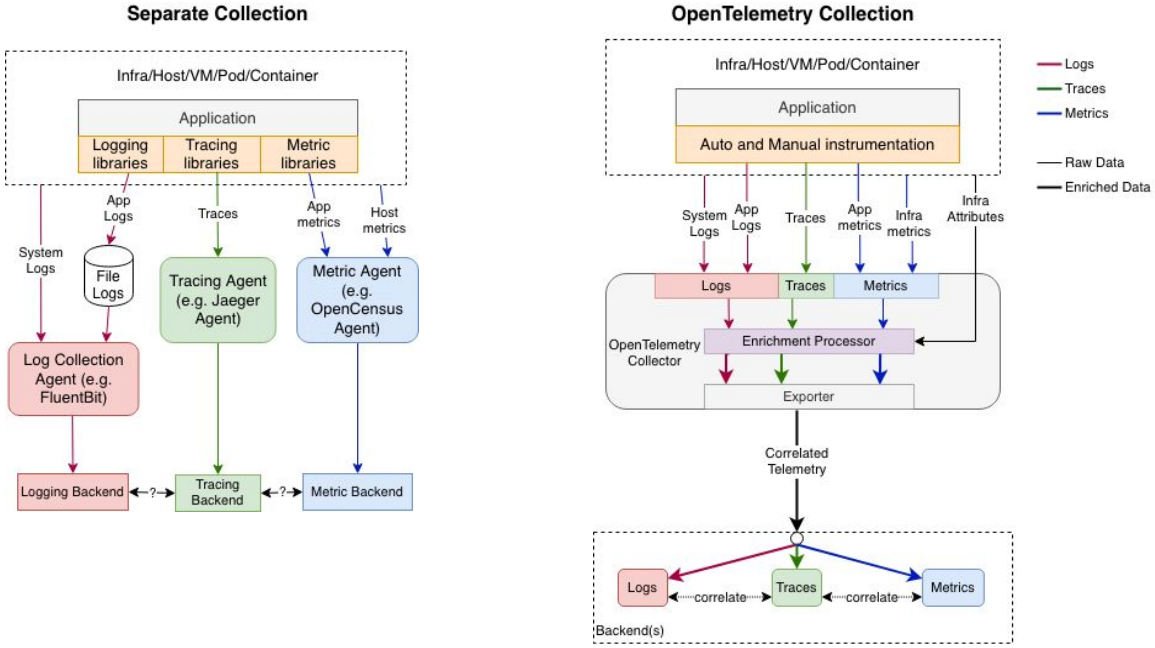
On distributed tracing, metrics, logs with Opentelemetry

Haixuan Xavier Tao

tao.xavier@outlook.com

Table of content

Opentelemetry Vision



Opentelemetry rust

Signal	Status
Logs	Alpha
Metrics	Alpha
Traces	Beta

Tracing


```
{
  "name": "hello",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x051581bf3cb55c13"
  },
  "parent_id": null,
  "start_time": "2022-04-29T18:52:58.114201Z",
  "end_time": "2022-04-29T18:52:58.114687Z",
  "attributes": {
    "http.route": "some_route1"
  },
  "events": [
    {
      "name": "Guten Tag!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```

```
{
  "name": "hello-salutations",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x93564f51e1abe1c2"
  },
  "parent_id": "0x051581bf3cb55c13",
  "start_time": "2022-04-29T18:52:58.114492Z",
  "end_time": "2022-04-29T18:52:58.114631Z",
  "attributes": {
    "http.route": "some_route3"
  },
  "events": [
    {
      "name": "hey there!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```

```
{
  "name": "hello-greetings",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x5fb397be34d26b51"
  },
  "parent_id": "0x051581bf3cb55c13",
  "start_time": "2022-04-29T18:52:58.114304Z",
  "end_time": "2022-04-29T22:52:58.114561Z",
  "attributes": {
    "http.route": "some_route2"
  },
  "events": [
    {
      "name": "hey there!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    },
    {
      "name": "bye now!",
      "timestamp": "2022-04-29T18:52:58.114585Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```

Logs data format

Span Context

```
{
  "name": "hello",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x051581bf3cb55c13"
  },
  "parent_id": null,
  "start_time": "2022-04-29T18:52:58.114201Z",
  "end_time": "2022-04-29T18:52:58.114687Z",
  "attributes": {
    "http.route": "some_route1"
  },
  "events": [
    {
      "name": "Guten Tag!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```

Logs

Field Name	Description
Timestamp	Time when the event occurred.
ObservedTimestamp	Time when the event was observed.
TraceId	Request trace id.
SpanId	Request span id.
TraceFlags	W3C trace flag.
SeverityText	The severity text (also known as log level).
SeverityNumber	Numerical value of the severity.
Body	The body of the log record.
Resource	Describes the source of the log.
InstrumentationScope	Describes the scope that emitted the log.
Attributes	Additional information about the event.

Implementation details

Super easy to get started in rust as `opentelemetry-rs` can directly subscribe to `tokio::tracing` with:

cargo add tracing

cargo add tracing-opentelemetry

Automatically export traces and logs into opentelemetry format.

tracing-opentelemetry

Logs

```
tracing::error!(“This is an error!”);

tracing::info!(“This is an info”);
```

Traces

```
#[tracing::instrument(level = "trace")]

let span = span!(Level::TRACE, "span", yaks);

let _enter = span.enter();
```

Cross language / Cross Process Context

Carrier (Inject/Extract)

KeyValue Store to store Trace-Id, Span-Id, Attributes …

Global TextMap Propagators

Set the values of the KeyValue Stores as Text

Opentelemetry Metrics

Universal data format for metrics data:

Some examples of use cases for metrics include:

- Reporting the total number of bytes read by a service, per protocol type.
- Reporting the total number of bytes read and the bytes per request.
- Reporting the duration of a system call.
- Reporting request sizes in order to determine a trend.
- Reporting CPU or memory usage of a process.
- Reporting average balance values from an account.
- Reporting current active requests being handled

Opentelemetry System Metrics

https://github.com/haixuanTao/opentelemetry-system-metrics

(Live Demo)

Opentelemetry System Metrics within dora-rs

(Live Demo)

Homework

– Check out: https://docs.rs/tracing-for-pyo3-logging/latest/tracing_for_pyo3_logging/fn.setup_logging.html

Or https://crates.io/crates/pyo3-pylogger

This can push python log event into dora log event which can be pushed into opentelemetry ❤️

Homework for metrics

You can try to add `nvidia GPU` data with `nvidia-smi` or NVML(https://docs.rs/nvml-wrapper/latest/nvml_wrapper/) into https://github.com/haixuanTao/opentelemetry-system-metrics that is going to make it easier to know GPU utilization.

Q&A Session