KUBERNETES COMMUNITY DAYS SHENZHEN 2023

让 Logs、Metrics 和 Traces 联动起来

eBPF 的上层应用

朱杰坤

趣丸科技

Software Engineer

OpenTelemetry Contributor

Agenda

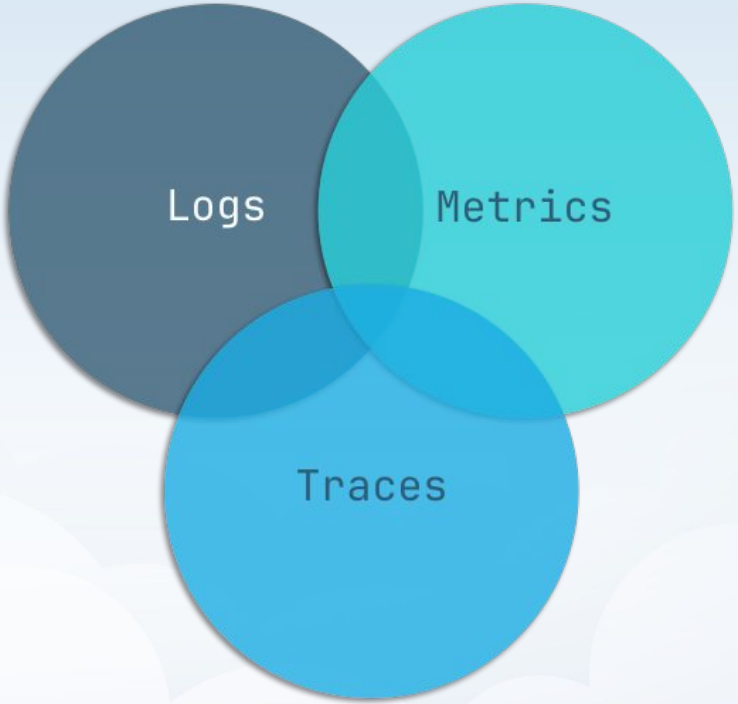
目录

Part 01

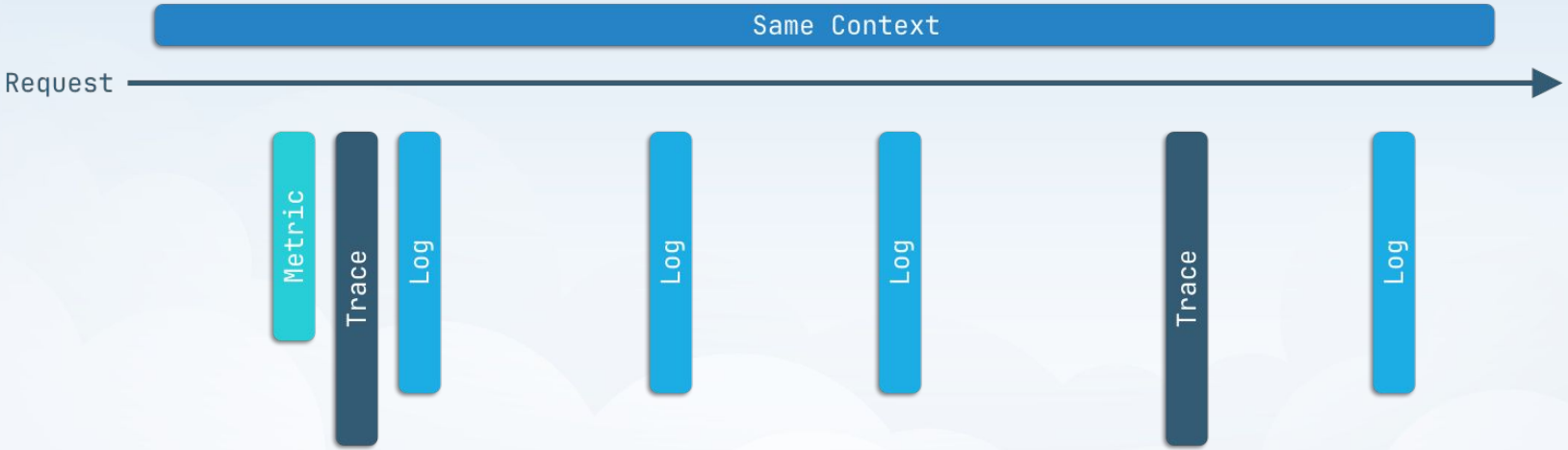
Connecting Everything

关联数据：Metrics, Logs and Traces

The Three Pillars of
Observability



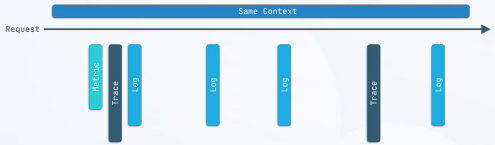
Context



exemplars

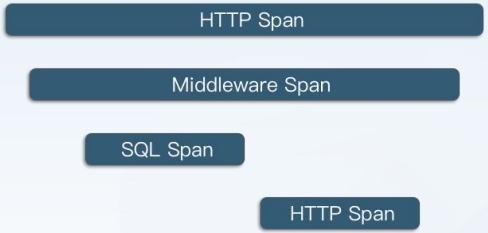
```
# TYPE http_request_duration_bucket histogram
http_request_duration_bucket{le="0.25"} 205    # {TraceID="938c2cc0dcc05f2b"} 0.1758 1.61519e+09
```

exemplars

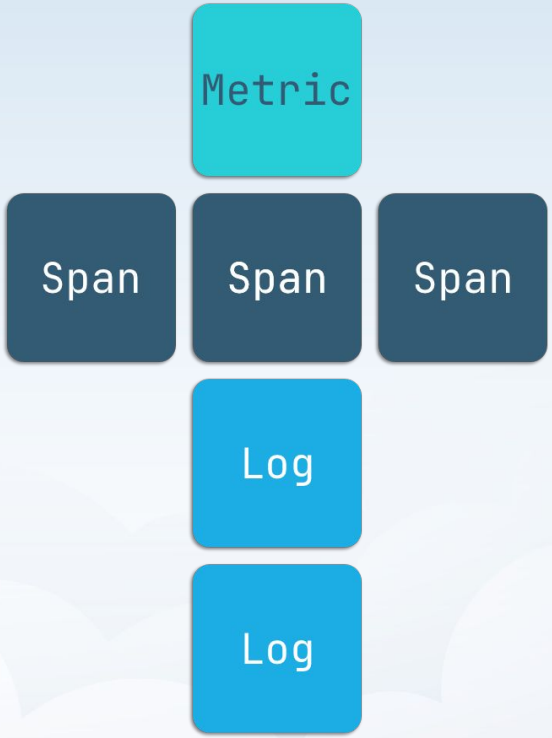



```
# TYPE http_request_duration_bucket histogram
http_request_duration_bucket{le="0.25"} 205 # {TraceID="938c2cc0dcc05f2b"} 0.1758 1.61519e+09
```

```
2023-12-02 21:35:03 [info]
938c2cc0dcc05f2b http access,
api="/user/info",param={"user_id":921
8,"status":1},body={},response={"user
name":"john","age":30}

2023-12-02 21:35:02 [info]
938c2cc0dcc05f2b http request,
url="https://downstream.com",body={"a
rg":1,"status":1},response={"age":30}

2023-12-02 21:35:01 [info]
938c2cc0dcc05f2b query database,
sql="select * from user where
user_id=9218"

2023-12-02 21:35:01 [info]
938c2cc0dcc05f2b safe check,result=0
```

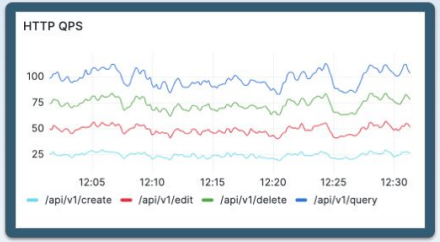


All-in-One Data Structure



Span Metrics Connector


```
{
  "trace_id": "0123456789abcdef0123456789abcdef",
  "span_id": "0123456789abcdef",
  "parent_span_id": "abcdef0123456789",
  "name": "HTTP Request",
  "kind": "CLIENT",
  "start_time": "2023-12-03T10:30:00Z",
  "end_time": "2023-12-03T10:31:00Z",
  "status": "OK",
  "attributes": {
    "http.method": "POST",
    "http.host": "https://example.com",
    "http.status_code": 200,
    "http.request.params": {
      "param1": "value1",
      "param2": "value2"
    },
    "http.request.body": "{\"key\": \"value\"}",
    "http.response.body": "{\"result\": \"success\"}",
    "service_name": "user_account",
    "endpoint_name": "/api/v1/query"
  }
}
```


```
2023-12-03 10:31:00, INFO, 0123456789abcdef0123456789abcdef, HTTP Request,
url=https://example.com/api/v1/query, method=POST, status=200,
param={"param1":"value1","param2":"value2"}, response={"result":"success"},
duration=60.00s
```

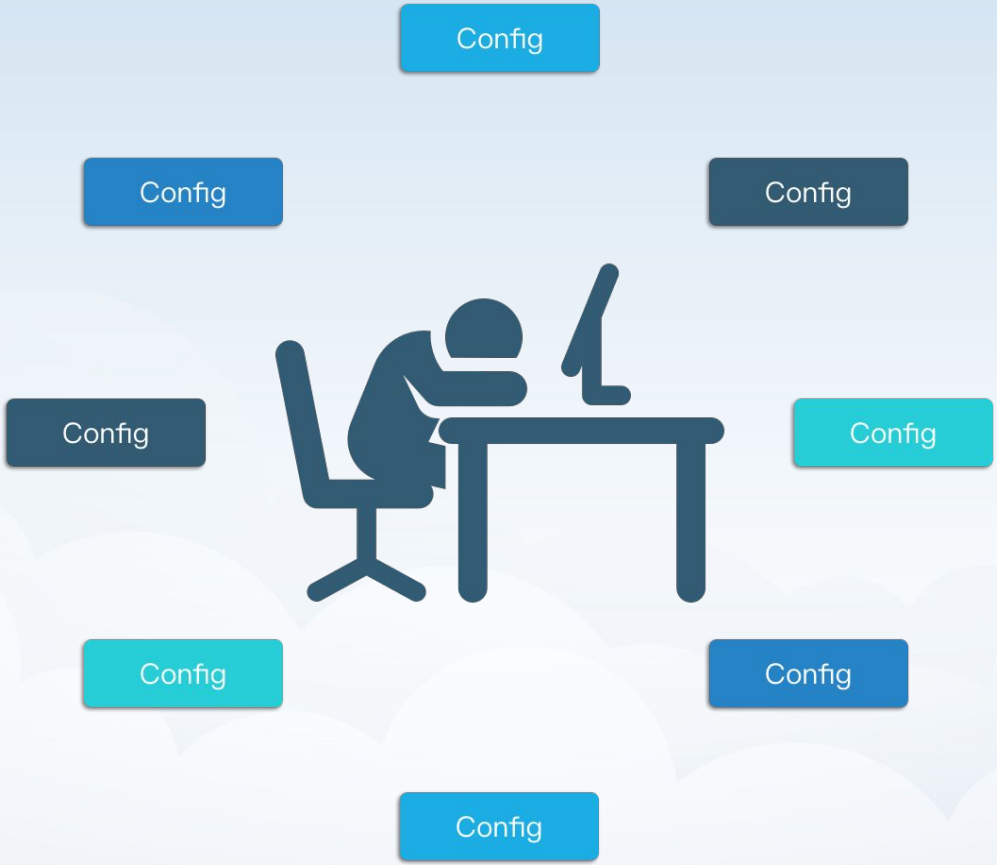
Part 02

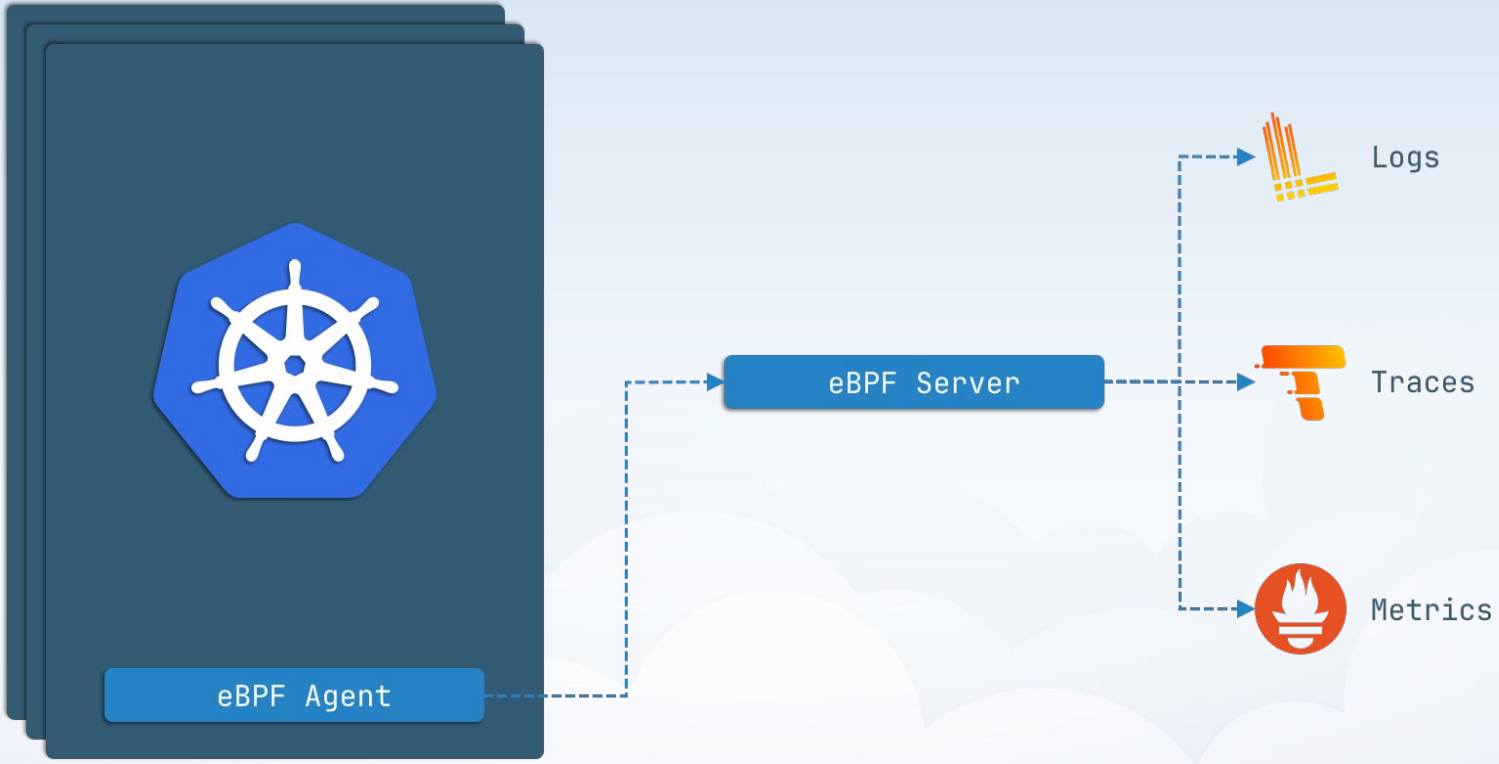
Why eBPF

埋点选择：为什么用 eBPF 代替 SDK

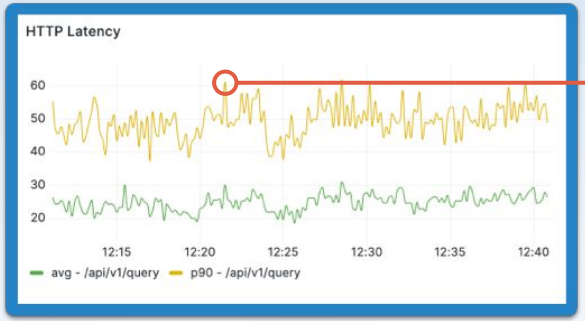
Diversity
of Tech Stacks

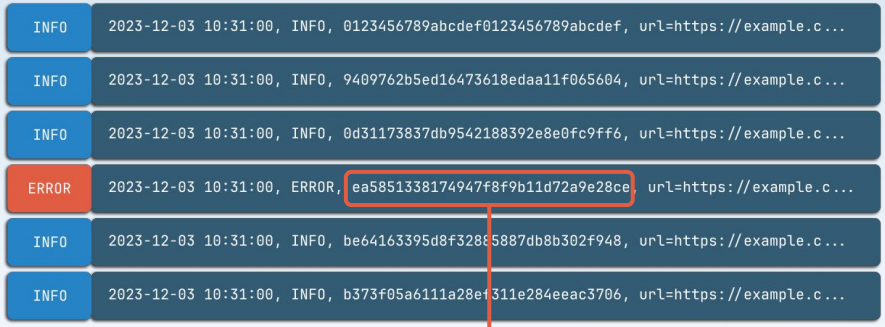
Frequently Asked
Questions

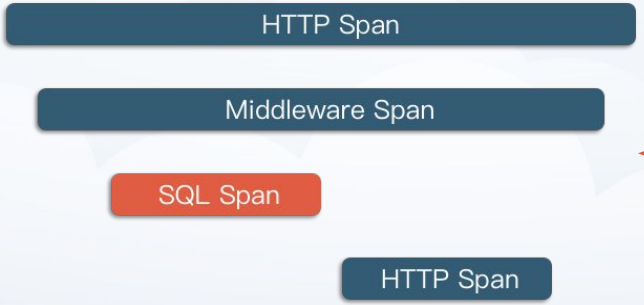

Config
Config
Config
Config
Config
Config
Config
Config


eBPF Agent
eBPF Server
Logs
Traces
Metrics

How Does It Look







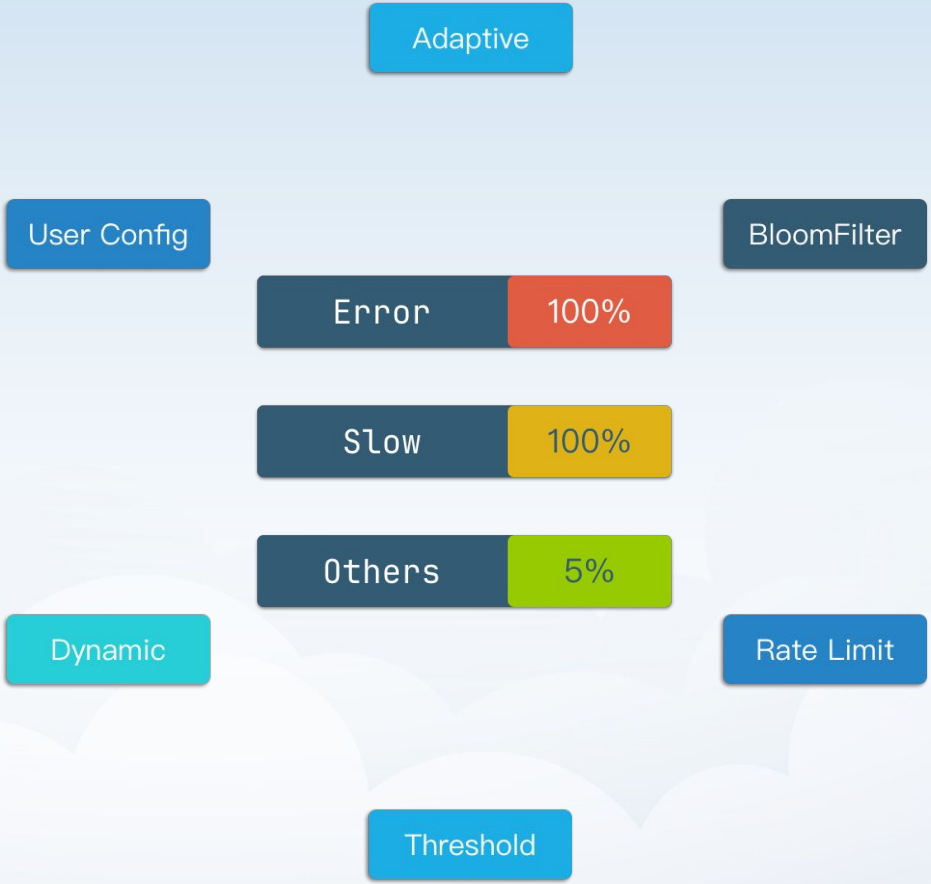
Part 03

Troubles Encountered with eBPF

采样、数据精度与 Trace 组装

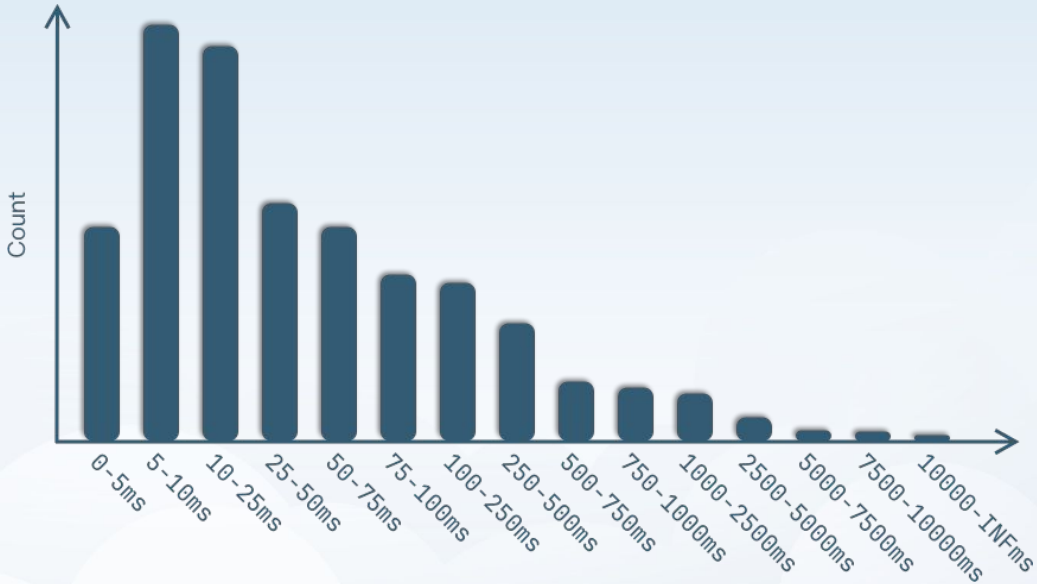
Reducing

Logs



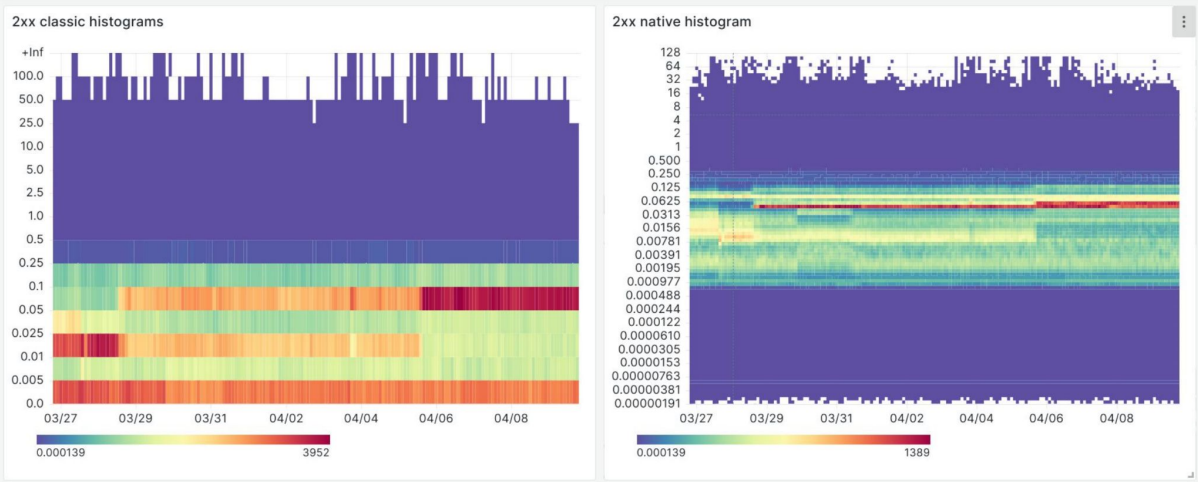
Histogram

for Diverse Scenarios



Options

Native Histogram



Spans

Without TraceID

Span (eBPF)

```
{
  "trace_id": "00000000000000000000000000000000",
  "span_id": "0000000000000000",
  "parent_span_id": "0000000000000000",
  "name": "HTTP Request",
  "kind": "CLIENT",
  "start_time": "2023-12-03T10:30:00Z",
  "end_time": "2023-12-03T10:31:00Z",
  "status": "OK",
  "attributes": {
    ...
```

Connecting eBPF Span

Span (SDK)

```
{
  "trace_id": "0123456789abcdef0123456789abcdef",
  "span_id": "0123456789abcdef",
  "parent_span_id": "abcdef0123456789",
  "name": "HTTP Request",
  "kind": "CLIENT",
  "start_time": "2023-12-03T10:30:00Z",
  "end_time": "2023-12-03T10:31:00Z",
  "status": "OK",
  "attributes": {
    ...
    "service_name": "user_account",
    "endpoint_name": "/api/v1/query"
  }
}
```

Span (eBPF)

```
{
  "trace_id": "00000000000000000000000000000000",
  "span_id": "0000000000000000",
  "parent_span_id": "0000000000000000",
  "name": "HTTP Request",
  "kind": "CLIENT",
  "start_time": "2023-12-03T10:30:00Z",
  "end_time": "2023-12-03T10:31:00Z",
  "status": "OK",
  "attributes": {
    ...
    "service_name": "user_account",
    "endpoint_name": "/api/v1/query",
    "tcp.req_seq": 123456789,
    "tcp.resp_seq": 987654321,
    "syscall.trace_id": "abcdef0123456789"
  }
}
```

DeepFlow

ACM SIGCOMM '23

查找关联 Span

排序

```
Input: start_span - User-Chosen Span, I - Iteration Times
Output: T - Assembled Trace
// Iterative Span Search
span_set ← Set(start_span)
filter ← {id == start_span.id}
for iter ← 1 to I do
  for s ← span_set do
    filter ← filter ∪ {systrace_id == s.systrace_id}
    filter ← filter ∪ {pseudo_th_id == s.pseudo_th_id}
    filter ← filter ∪ {x_req_id == s.x_req_id}
    filter ← filter ∪ {tcp_seq == s.tcp_seq}
    filter ← filter ∪ {trace_id == s.trace_id}
  end for
  span_set = search_database(filter)
  if span_set.not_update then
    Break
  end if
end for
// Set Parent for Each Span
for s ← span_set do
  for r_s ← s.related_spans() do
    if related_s.is_parent(s) then
      s.set_parent(related_s)
    end if
  end for
end for
T ← span_set.sort()
Return T
```

Part 04

Conclusion

总结

No Silver Bullet

没有银弹

Thank You