Polytechnique Montréal - January 2022

LTTng and Related Projects Updates

EfficiOS

Outline

- LTTng 2.13
  - Event-rule matches triggers
  - LTTng-UST 2.13: Application rebuild required
- LTTng 2.14 (ongoing development)
  - Aggregation Maps and Trace Hit Counters
- LTTng 2.15 and Babeltrace 2.1 (ongoing development)
  - Common Trace Format 2 (CTF 2)
- Restartable Sequences

EfficiOS

LTTng 2.13: Event-rule Matches Trigger

Original requirement:

- Use the snapshot tracing mode on a group of machines
- Application emits an event when this problem occurs
  - Event record contains information on the other machines
- Use this event’s payload to quickly record a snapshot on the other machines

This is a pretty complex requirement with many moving parts

EfficiOS

Large Extension of the Trigger Mechanisms

- Requirement fits into the use case of triggers, introduced in 2.10
  - A trigger associates a condition to one or more actions
- Original use-case for triggers
  - Monitor the usage level of buffers
  - Notify a control application which enables/disables event-rules
- Triggers were expanded for LTTng 2.11
  - Can monitor for completed recording session rotations
  - Notify an application to process the archived trace chunk

Event-rule Matches Condition

- A new condition type is needed
  - Event-rule matches
    - Name, domain, filter expression
  - First condition that involves the tracers

- The rest of the trigger infrastructure can take action when this condition is met
  - Not intended as a replacement for the existing low-overhead tracing facilities
  - React to an event without needing to consume the trace itself

New Actions

- Existing *notify* action
  - Is the most flexible action mechanism
  - Requires the development and deployment of a client to receive notifications
  - Too complicated for simple actions

- New actions
  - Starts or stop a recording session
  - Rotate a recording session
  - Take a snapshot of a recording session
  - Groups of actions

EfficiOS

The pieces of the puzzle start fitting together

- When monitoring a system
  - Register a trigger
    - Event-rule matches condition targeting the event of interest
    - Associated to an action (*record a snapshot*)

- Some pieces are still missing
  - What about recording a snapshot on *other machines*?

Event Payload Capture

- Event-rule matches condition
  - Allows a *capture descriptor* to be specified
    - Capture event record or context field
    - Associated to an action (*record a snapshot*)
  - Captured payloads are made available to actions
  - Can be transmitted as part of a notification to an external application

EfficiOS

The Complete Story

- Register a trigger that will
  - Take a snapshot whenever a specific event occurs
  - Notify an external application with a subset of the event’s payload
- Captured payloads are made available to actions
  - Can be transmitted as part of a notification to an external application
- These enhanced trigger facilities are now also available using the `lttng` command line client

EfficiOS

LTTng-UST 2.13: Application Rebuild Required

- LTTng-UST 2.13 introduces a liblttng-ust soname major version bump,
- Users **must recompile** their instrumented applications/libraries and tracepoint provider packages to use LTTng-UST 2.13,
- This change became a necessity to clean up the library and for liblttng-ust to stop exporting private symbols,
- The LTTng-UST 2.12 instrumentation API is still available through a compatibility layer (enabled at compilation by default),
- Notable change: LTTng-UST now only depends on liburcu at build time, not at run time.

LTTng 2.14 (Ongoing Development)

- New in LTTng 2.14: Aggregation Maps and Trace Hit Counters

Tracing vs Aggregation (1/3)

Tracing

```
[18:21:19.648266565] (+0.001025307) raton my_app:adjust_sensor: { cpu_id = 1 }, { id = 3 }
[18:21:19.648278383] (+0.000001329) raton my_app:curr_temp: { cpu_id = 1 } { temp = 53, status = OK }
[18:21:19.648277054] (+0.000010489) raton my_app:empty: { cpu_id = 2 }, { }
[18:21:19.648278948] (+0.000000565) raton my_app:curr_temp: { cpu_id = 5 }, { temp = 64, status = OK }
[18:21:19.648279875] (+0.000000317) raton my_app:curr_temp: { cpu_id = 1 }, { temp = 98, status = OK }
[18:21:19.648283004] (+0.000000571) raton my_app:temp_too_high: { cpu_id = 1 }, { temp = 103, status = OVERHEATING }
```

Tracing vs Aggregation (2/3)

Aggregation

```
+------------------------+--------+
|          name          | count  |
+------------------------+--------+
| my_app:adjust_sensor   |      6 |
| my_app:curr_temp       |     53 |
| my_app:temp_too_high   |      1 |
+------------------------+--------+
```

EfficiOS

Tracing vs Aggregation (3/3)

Tracing

- Event ordering
- Precise timing
- Payload recording

Aggregation

- Event counting
- Event grouping
- High level view

Concrete examples (1/2)

- Report the number of times an event occurred

```
+----------+--------+
|   name   | count  |
+----------+--------+
| event_1  |    571 |
| event_2  |   4163 |
| event_3  |      7 |
+----------+--------+
```

EfficiOS

Concrete examples (2/2)

- Report event occurrence by subsystems

```
+----------------+-------+
|      name      | count |
+----------------+-------+
| data_thread    |   853 |
| ui_thread      |   190 |
| control_thread |  5621 |
+----------------+-------+
```

Maps

- Maps are key-value stores
  - `string -> signed integer`
  - are part of tracing sessions
- Configuration options:
  - Domain,
  - Buffer type,
  - Bucket size,
  - Number of buckets.

Trace Hit Counter

- Similar to regular LTTng events,
- Apply on a specific session and map,
- **Arbitrary key,**
- Exposed through the LTTng Trigger interface,
  - `on-event` condition,
  - One or more `incr-value` actions.

EfficiOS

CLI - Typical Workflow

- **Create a `on-event` and `incr-value` trigger**
- Create session
- **Create map**
- Start session
- Run workload
- Stop session
- **Visualize the map**

CLI - incr-value action

```
$ lttng add-trigger \
        --condition on-event --userspace "tp:*" \
        --action incr-value \
         --session my_session \
         --map my_map \
         --key 'my_incr_value_${EVENT_NAME}'
```

Arbitrary keys created using the key syntax:

- Literal string,
- Event (name or provider).

Examples:

- `--key "Event category #2"`
- `--key "${EVENT_NAME}_postfix"`

EfficiOS

CLI - add-map

Maps offer multiple configuration options:

```
$ lttng add-map \
    --userspace \
    --session mysession \
    --per-uid \
    --bitness 32 \
    --max-key-count 4096 \
    mymap
```

EfficiOS

CLI - `list`

Maps are listed in the existing `list` and `status` commands:

```
$ lttng status
```

Or

```
$ lttng list my_session
[...]
Maps:
-------------
- my_map (enabled)
       Attributes:
              Bitness: 32
              Counter type: per-uid
              Boundary policy: OVERFLOW
              Bucket count: 4096
              Coalesces hits: TRUE
```

CLI - `view-map`

The content of a map can be viewed using the `view-map` command.

Shows the **value**, the **underflow**(uf) and **overflow**(of) flags for each key.

```
$ lttng view-map my_map

Session: 'my_session', map: 'my_map', map bitness: 64
UID: 1000, CPU: ALL
+-------------------+-----+----+----+
| key               | val | uf | of |
+-------------------+-----+----+----+
| Event category #2 |  20 |  0 |  0 |
+-------------------+-----+----+----+
| tp_tptest1        |  10 |  0 |  0 |
+-------------------+-----+----+----+
| tp_tptest5        |  10 |  0 |  0 |
+-------------------+-----+----+----+
| tptest1:postfix   |  10 |  0 |  0 |
+-------------------+-----+----+----+
| tptest5:postfix   |  10 |  0 |  0 |
+-------------------+-----+----+----+
```

EfficiOS

CLI - `view-map`

The value of a specific key can accessed using the `--key` option:

```
$ lttng view-map my_map --key 'tptest1:postfix'

Session: 'my_session', map: 'my_map', map bitness: 64
UID: 1000, CPU: ALL
+-----------------+-----+----+----+
| key             | val | uf | of |
+-----------------+-----+----+----+
| tptest1:postfix |  10 |  0 |  0 |
+-----------------+-----+----+----+
```

EfficiOS

Future avenues

- `decr-value` action,
  - Decrement the value of a map bucket,
  - Account entry and exit of functions or syscalls,
- Aggregate based on event payload fields,
- Increment based on event payload fields.
- Ring buffer usage accounting mode,
  - Estimate memory needed of a tracing workload,
  - Based on event occurrence and size.

Summary - Aggregation Maps

Aggregation allows for cheap and quick overview and analysis.

Aggregation is useful to tune tracing configuration for a given workload.

Aggregation allows for easy extraction of metrics.

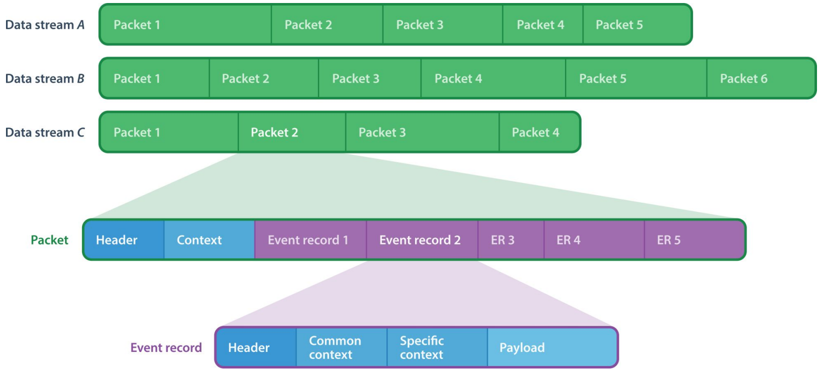
Common Trace Format (CTF)

- "<u>C</u>ommon <u>T</u>race <u>F</u>ormat"
- Self-described binary trace format
- CTF 1 specified in 2010-2011
- Focused on producer's performance, supporting:
  - Big-endian and little-endian fields
  - Bit fields
  - Custom field alignments
  - Multiple data streams

EfficiOS

Anatomy of a CTF Trace

- One metadata stream
- Zero or more data streams
- Zero or more auxiliary streams (**new to CTF 2**)



EfficiOS

Limitations of CTF 1: Summary

- Metadata language is hard to **consume**
- Metadata language is hard to **extend**
- Missing useful/needed **field types**:
  - Bit array
  - Variable-length integer
  - Boolean
  - Optional
  - BLOB
- Hard to attach data to a **specific trace**

EfficiOS

Common Trace Format 2 (CTF 2) Timeline

Date	Event
October 25, 2016	First specification proposal
November 18, 2016	DiaMon conference call about CTF2
October 27, 2017	“Introduction to CTF2” talk @ Tracing Summit
November 18, 2020	Second specification proposal
November 25, 2021	First specification release candidate
December 9, 2021	Second specification release candidate
December 17, 2021	Third specification release candidate

CTF 2: What’s New ?

- Trace metadata now expressed as JSON rather than custom DSL,
- Require explicit references and descriptions to simplify trace consumers,
- Remove type aliases (not much used in CTF 1),
- Keep semantic compatibility with TSDL:
    - A tracer producing a CTF 1.8 data stream can move to CTF 2 just by changing the metadata format.

CTF 2: What’s New ? (2)

- Introduce user-attributes property in selected metadata objects:
  - Field classes, event record classes, data stream classes, trace class, and the rest.
- User attributes are part of a specific namespace (trace vendor, specification, etc) to avoid conflicts.

CTF 2: What’s New ? (3)

- Introduce new field types:
  - Fixed-length bit array field class,
  - Variable-length integer and enumeration field classes:
    - Use LEB128 encoding.
  - Fixed-length boolean field class,
  - “Optional” field class,
    - Optional field dynamically enabled by a boolean/integer selector field,
    - Occupies 0 data stream bits if disabled.
  - Static-length and dynamic-length BLOB field classes:
    - Similar to array field classes, but with more constraints,
    - Has an IANA media type (MIME).

CTF 2: What’s New ? (4)

- Introduce optional **auxiliary streams** to contain trace-specific data,
- Example: The specific environment of the trace (TSDL *env* block),
- An auxiliary stream uses JSON.

CTF 2: Planned Adoption

- Babeltrace (source and sink): v2.1
- LTTng: v2.15
- barectf: as needed
- Trace Compass: EfficiOS collaborates with the Ericsson Trace Compass team to ensure timely CTF 2 support.

Restartable Sequences

- Linux kernel **rseq** system call merged in Linux 4.18 (in August 2018),
- Support for restartable sequences merged into glibc in December 2021:
    - Release 2.35 planned for February 2022,
- Will eventually enable fast per-cpu data accesses:
    - LTTng-UST ring buffer
    - LTTng-UST aggregation maps
    - Memory allocators (tcmalloc, jemalloc, libc malloc)
- Working on a *librseq* library to provide rseq support for applications linked against older glibc.

Resources

- LTTng project: https://lttng.org
- CTF website: https://diamon.org/ctf/
- CTF 2 specification RC:
  - http://diamon.org/ctf/files/CTF2-SPECRC-3.0.html
- EfficiOS blog post:
  - *"The 5-year journey to bring restartable sequences to Linux"*
  - https://www.efficios.com/blog/2019/02/08/linux-restartable-sequences/

EfficiOS