Polytechnique Montréal – May 2019

What’s new at EfficiOS ?

EfficiOS

mathieu.desnoyers@efficios.com

Outline

- Linux kernel and glibc contributions,
- LTTng,
- Babeltrace 2.

Linux Kernel and glibc Contributions

- Restartable sequence (rseq):
  - glibc: patch to automatically register rseq TLS for each thread ready, should be merged soon,
    - ABI between libraries to coordinate rseq registration,
    - discussion with architecture maintainers to choose code signature before each abort handler,
    - glibc use-case: use rseq to speed up sched_getcpu(3).
  - librseq,
- CPU operation vectors (cpu_opv):
  - Linus requests examples of rseq real-life users before merging additional code,
  - Required for using rseq from LTTng-UST.

LTTng 2.11

- Currently at 2.11-rc1 (release candidate 1),
- (Still) Fixing session rotation corner-cases and documentation before final version,
  - Taking much longer than expected:
    - Required changes have impact on protocol between session daemon and relay daemon,
      - preventing us from releasing without those changes,
    - Require design changes in the object model within relay daemon,
- Expect 2.11 final end of May,
- New features:
  - Session rotation,
  - Dynamic instrumentation with uprobes,
  - Filtering on array and sequence integers in LTTng-UST and LTTng-modules.
  - Filtering: bitwise operators,
  - Kernel tracer: kernel and user-space callstack contexts.

LTTng 2.11 – Session Rotation

- Split trace in self-contained traces on the fly,
- Allow processing portion of the trace without stopping tracing,
- Allows for pipelining and/or sharding of analyses (scale-out distributed analysis),
- Encryption, compression, cleanup of old chunks, integration with external message bus tools,
- *Fine-grained Distributed Application Monitoring Using LTTng*, Jérémie Galarneau, Open Source Summit 2018.

LTTng 2.11 – Dynamic instrumentation with uprobes

- Adding tracepoints without having to recompile or restart a process,
- Using the uprobe interface,
- Tracing userspace using the **kernel tracer**,
- Supported instrumentation point types:
  - ELF symbols,
  - SystemTap/SDT probe points (without semaphore).

```
lttng enable-event --kernel
  --userspace-probe=elf:/path/to/binary:symbol
    event_name
```

EfficiOS

LTTng 2.11 – Dynamic instrumentation with uprobes

- Limitations:
  - Slower than LTTng-UST, because of context-switches to the kernel,
  - No tracepoint payload recorded at the moment.

Filtering on array and sequence of integers

- Filter out event based on the content of arrays and sequence

```
[14:32:57.03] host lttng_ust_prov:event : { _field_length = 4,
field = [ [0] = 121, [1] = 55, [2] = 23, [3] = 42 ] }
```

- Define filter using indexes in sequence:

```
lttng enable-event --userspace lttng_ust_prov:event
          --filter='field[0]<100 && field[3]==42'
```

EfficiOS

Filtering: Bitwise Operators

- Support bitwise operators in both kernel and user-space tracers:
  - Bitwise NOT (~),
  - Bitwise left/right shift (<</>>),
  - Bitwise AND (&),
  - Bitwise OR (|),
  - Bitwise XOR (^).

Kernel and User-Space Callstack Contexts

- In lttng-modules kernel tracer,
- Sample kernel and user-space callstacks as a context,
- Main use-case: sample user-space callstack on system call entry,
- Requires applications and libraries to be built with frame pointers to unroll user-space stacks.

Upcoming LTTng 2.12 Features

- User ID tracker,
- Relay daemon enhancements:
  - Categorize trace hierarchy by session / hostname,
  - Allow overriding current working directory,
  - LRU tracking of open file descriptors.
- Fast LTTng clear,
- LTTng container support,
  - Namespace hierarchy,
  - Namespace event contexts,
- LTTng 2.12-rc1 time-line will be planned when 2.11 final is released.

Ongoing Work: 2019 (LTTng 2.13)

- LTTng dynamic snapshot and event notification:
  - Sampling counters per trigger,
  - Send partial event payload with trigger notification,
  - Reporting of discarded trigger messages due to overload,
- LTTng strace-alike follow children:
  - Trace a hierarchy of processes with the PID tracker.
- Trace hit counters per tracepoint.
- LTTng containers: Docker support.

Babeltrace 2.0-pre5

- Merged back ~1.5 year of work into babeltrace master branch as of 2.0-pre5,
- Babeltrace 2.0-pre5 (May 3, 2019):
  - Performance improvements,
    - Improve trace processing graph implementation for speed,
    - Pre-allocate and re-use objects,
    - Avoid runtime cost at every field/event read,
    - Update public APIs,
    - Developer mode,
    - Loglevel tuning,
  - Supports trace merging from LTTng session rotation archives,
  - LTTng live “subscribe” mode.

Towards Babeltrace 2.0

- API nearly stabilized.
- Currently a team of 3 full time developers working towards Babeltrace 2.0,
- Planned:
  - Stabilize API (rc1),
  - Co-installation with Babeltrace 1.x,
  - Windows support,
  - Documentation,
  - Miscellaneous fixes.