Tracing Summit 2018

LTTng Status Update 2018





Linux Kernel: Membarrier System Call

- Membarrier is a system call issuing a memory barrier on a set of threads,
- Provide LTTng-UST ring buffer and liburcu read-side performance enhancement,
- New commands:
 - Private expedited (4.14),
 - Shared renamed to Global (4.16),
 - Global expedited (4.16),
 - Private expedited sync core for JIT reclaim (4.16).



Linux Kernel: Restartable Sequences

- Restartable Sequences (rseq) is a newly proposed system call which accelerates user-space operations on per-cpu data.
- Amongst its use-cases, will improve LTTng-UST ring buffer performance when integrated into LTTng-UST:

```
* LTTng-UST: write event 32-bit header, 32-bit payload into tracer per-cpu buffer
```

```
getcpu+atomic (ns/op) rseq (ns/op) speedup arm32: 2502.0 2250.0 1.1 x86-64: 117.4 98.0 1.2
```



Restartable Sequences Upstreaming Status

- Linux 4.18:
 - rseq system call merged,
 - rseq wired up for x86 32/64, powerpc 32/64, arm 32, mips 32/64,
- Linux 4.19:
 - rseq wired up for arm 64, s390 32/64,
- Ongoing work:
 - librseq,
 - glibc rseq registration/unregistration at thread start/exit,
 - new cpu opv system call.



LTTng 2.11

New features:

- Session rotation,
- Dynamic instrumentation,
- Filtering on array and sequence integers in LTTng-UST and LTTngmodules.
- 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 of 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

- 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
```



LTTng 2.11 – Dynamic instrumentation

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'</pre>
```



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,
- Fast LTTng clear,
- Relay daemon enhancements
 - Categorize trace hierarchy by session / hostname,
 - Allow overriding current working directory,
 - LRU tracking of open file descriptors.



Planned for 2019

LTTng dynamic snapshot and event notification.

