libside: Giving the Preprocessor a Break with a Tracer-Agnostic Instrumentation API

Mathieu Desnoyers, EfficiOS

Linux Plumbers Conference
November 14, 2023
Richmond, VA

EfficiOS

Outline

- User events,
- Userspace Instrumentation Desiderata,
- LTTng-UST Tracepoints,
- Libside
- Future Work

User Events

- Exposes a stable ABI allowing applications to register their event names/field types to the kernel,
- Can be expected to have a large effect on application instrumentation,
- My concerns:
  - Should be co-designed with a userspace instrumentation API/ABI rather than only focusing on the kernel ABI,
  - Should allow purely userspace tracers to use the same instrumentation as userspace tracers implemented within the Linux kernel,
  - Tracers can target their specific use-cases, but infrastructure should be shared,
  - Limit fragmentation of the instrumentation ecosystem.

Userspace Instrumentation Desiderata

- Common instrumentation for kernel and purely userspace tracers,
- Instrumentation is self-described,
- Support compound and nested types,
- Support pre-registration of events,
- Do not rely on compiled event-specific code,
- Independent from ELF,
- Simple ABI for instrumented code, kernel, and user-space tracers,
- Support concurrent tracers,
- Natively cover statically-typed and dynamically-typed languages:
  - C/C++, Golang, Java, .NET, Python, Javascript, Rust, Erlang, and other runtimes,
- Expose API to allow dynamic instrumentation libraries to register their events/payloads.

LTTng-UST Tracepoints

- Based on compilation of tracepoint probes,
- Multi-pass header inclusion:
  - hard to understand compiler errors.
- Instruction cache pollution when tracing is active,
- Only supports static instrumentation for C/C++,
- No support of nested compound types,
- Supports a single tracer (LTTng-UST),
- Static type checking between call sites and probe signature done at compile-time.

libside

- **S**oftware **I**nstrumentation **D**ynamically **E**nabled,
- https://github.com/efficios/libside
- Instrumentation API/ABI:
  - Type system,
  - Helper macros for C/C++,
  - Express instrumentation description as data,
  - Instrumentation arguments are passed on the stack as a data array (similar to iovec) along with a reference to instrumentation description,
  - Instrumentation is conditionally enabled when at least one tracer is registered to it.
- Tracer-agnostic API/ABI:
  - Available events notifications,
  - Conditionally enabling instrumentation,
  - Synchronize registered user-space tracer callbacks with RCU,
  - Co-designed to interact with User Events.

Instrumentation Helper Macros

```
#include <side/trace.h>

side_static_event(my_provider_event, "myprovider", "myevent",
        SIDE_LOGLEVEL_DEBUG,
        side_field_list(side_field_s32("myfield"))
);

int main()
{
        side_event(my_provider_event, side_arg_list(side_arg_s32(42)));
        return 0;
}
```

Integer Field Base Attribute Example

- Integer field:

side_field_u16("u16base2", side_attr_list(side_attr("std.integer.base", side_attr_u8(2))))

- Prints as:

u16base2: { attr: [{ key: "std.integer.base", value: 2 }], value: 0b0000000000110111 }

How Tracers Interact with libside

- User-space tracer
  - Register callback to be notified when a new event description is available,
  - Register callback to be called when an event is emitted.
- Kernel tracer
  - Libside invokes User Events ioctls(),
  - User Events modifies the enabled state when it wishes to be called when a libside event is emitted.

Future Work

- Extensibility of libside ABI,
- Validate description vs payload type match:
  - Runtime checker,
  - Static analyzer,
- Application state dump,
- Integration with LTTng-UST:
  - Implement event registration notification callbacks,
  - Bytecode interpreter,
- Integration with User Events:
  - Event fields types description ABI,
  - Event fields payload content ABI.

Questions / Comments ?

- For more information:
  - Tracing Summit 2023
  - https://tracingsummit.org/ts/2023/libside/
    - Slides:https://tracingsummit.org/ts/2023/files/Tracing-Summit-2023-libside.pdf
    - Video: https://youtu.be/35G4rbf58uY