

# LTTng base address recording prototype

## Current Status:

Submitted to lttng-dev mailinglist. Pending for Review/Discussion.

## Design:

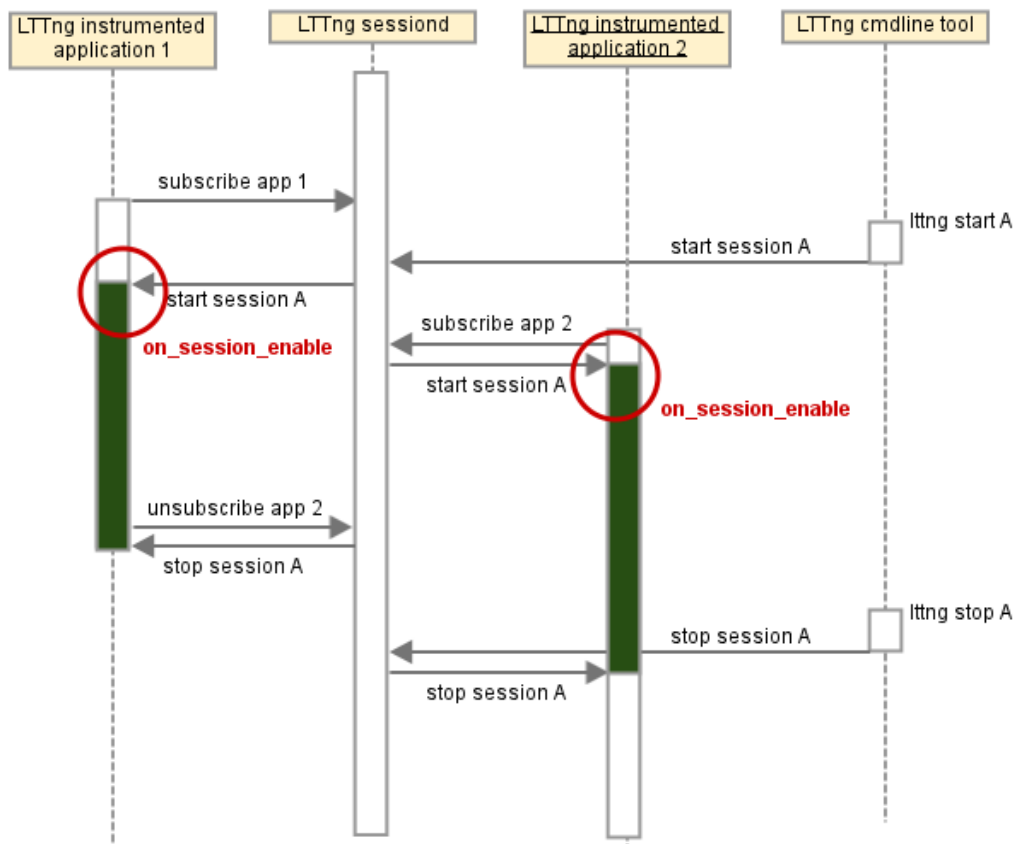
Several hooks have been identified in the lttng-ust framework that are used for the implementation of base address recording.

### Hook 1: Session daemon listener thread: receive session start

Whenever an instrumented **application receives a "session start" message**, all currently **active shared objects** are iterated and **recorded as ust\_baddr:init** events.

This solution allows us to get rid of our met\_soinfo\_dump utility from mettools.

## Hooking into LTTng session handling



On lttng-dev I suggested making that hook accessible to other lttng users as well:

*Ideally this hook should be provided as API functionality, controllable via lttng commands. For example:*

```
lttng on-session-enable on-session-enable_state_dumper.so
```

*The requirement of dumping state information on session\_enable is generic enough that other instrumentation could*

also benefit from it (e.g. *stack dump for libltng-ust-cyg-profile/libltng-ust-cyg-profile-fast*).

### **Hook 2-2: Static ctor and dtor in ltng/tracepoint.h**

For an application to use LTTng tracepoints the user **always** has to include a respective tracepoint provider header file.

For example:

```
#define TRACEPOINT_DEFINE
#include "test_tracepoint.h"

void foo()
{
    tracepoint( test, info, "foo is called" );
}
```

**Each of this tracepoint provider header file includes ltng/tracepoint.h.** ltng/tracepoint.h contains static ctors and dtors that take care the tracepoints (of the tracepoint provider header) are prepared and registered to the LTTng framework. **The patch adds the following behavior to ltng/tracepoint.h:**

- Emit an **ust\_baddr:push** event after tracepoint registration that records base address, path, size and last modification time of shared object/executable.
- Emit an **ust\_baddr:pop** event before tracepoint deregistration that records base address (of the object that is about to get closed).

The ctors and dtors in ltng/tracepoint.h are constructed in a way that prevents recording more than one push/pop pair per shared object/executable even if several tracepoint provider header files are used in a shared object/executable.

### **Details and example traces:**

**[ltng-dev] [RFC PATCH ltng-ust] Make ltng-ust aware of shared object base addresses (issue #474)**

<http://lists.lttng.org/pipermail/ltng-dev/2013-August/021264.html>