

LTTng-UST - Bug #1257

lttng-ust-java-tests failure on latest master

04/20/2020 03:39 PM - Michael Jeanson

Status:	Resolved	Start date:	04/20/2020
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

Since the introduction of 'Refactor representation of nested types' in the master branch of ust and tools, the java agent test suite fails at 'org.lttng.ust.agent.integration.context.JulAppContextIT#testContextValueFloat'.

The output of the test suite shows the JVM dying on a double free ([Full log](#)) :

```
06:46:43 Running org.lttng.ust.agent.integration.context.JulAppContextIT#testContextValueFloat
06:46:43 $ lttng create 8096ae8b-c7bb-4255-9664-924a62fb4948
06:46:43 Session 8096ae8b-c7bb-4255-9664-924a62fb4948 created.
06:46:43 Traces will be output to /home/jenkins/lttng-traces/8096ae8b-c7bb-4255-9664-924a62fb4948-20200419-104643
06:46:43 (returned from command)
06:46:43 AbstractLttngAgent: Initializing Agent for domain: JUL
06:46:43 LttngTcpSessiondClient: (root=true, domain=3) Connecting to sessiond
06:46:43 LttngTcpSessiondClient: (root=false, domain=3) Connecting to sessiond
06:46:43 LttngTcpSessiondClient: (root=false, domain=3) Registering to sessiond
06:46:43 LttngTcpSessiondClient: (root=false, domain=3) Waiting on sessiond commands...
06:46:43 LttngTcpSessiondClient: (root=false, domain=3) Registration done
06:46:46 LttngTcpSessiondClient: (root=true, domain=3) Connecting to sessiond
06:46:46 $ lttng enable-event -j -a -s 8096ae8b-c7bb-4255-9664-924a62fb4948
06:46:46 LttngTcpSessiondClient: (root=false, domain=3) Received enable event command: SessiondEnableEventCommand[eventName=*, logLevel=LTNG_EVENT_LOGLEVEL_ALL, filterString=]
06:46:46 LttngTcpSessiondClient: (root=false, domain=3) Sending response: LttngAgentResponse[code=1, success]
06:46:46 All JUL events are enabled in channel channel0
06:46:46 (returned from command)
06:46:46 $ lttng add-context -j -t $app.Retriever1:some.context_name -s 8096ae8b-c7bb-4255-9664-924a62fb4948
06:46:46 LttngTcpSessiondClient: (root=false, domain=3) Received enable app-context command
06:46:46 LttngTcpSessiondClient: (root=false, domain=3) Sending response: LttngAgentResponse[code=1, success]
06:46:46 JUL context $app.Retriever1:some.context_name added to all channels
06:46:46 (returned from command)
06:46:46 $ lttng start 8096ae8b-c7bb-4255-9664-924a62fb4948
06:46:46 libust[1265686/1265707]: Message Received "Create Session" (64), Handle "root" (0) (in print_cmd() at lttng-ust-comm.c:467)
06:46:46 libust[1265686/1265707]: Return value: 1 (in handle_message() at lttng-ust-comm.c:1085)
06:46:46 libust[1265686/1265707]: message successfully sent (in send_reply() at lttng-ust-comm.c:643)
06:46:46 libust[1265686/1265707]: Message Received "Create Channel" (81), Handle "session" (1) (in print_cmd() at lttng-ust-comm.c:467)
...
06:46:46 libust[1265686/1265707]: Message Received "Enable" (128), Handle "session" (1) (in print_cmd() at lttng-ust-comm.c:467)
06:46:46 libust[1265686/1265707]: Sent register event notification for name "lttng_jul:event": ret_code 0, event_id 0
06:46:46 (in ustcomm_register_event() at lttng-ust-comm.c:1517)
06:46:46 liblttng_ust_tracepoint[1265686/1265707]: Registering probe to tracepoint lttng_jul:event. Queuing release. (in __tracepoint_probe_register_queue_release() at tracepoint.c:652)
06:46:46 liblttng_ust_tracepoint[1265686/1265707]: Release queue of unregistered tracepoint probes
```

```
. (in __tracepoint_probe_prune_release_queue() at tracepoint.c:743)
06:46:46 libust[1265686/1265707]: Sent register enum notification for name "dynamic_type_enum": re
t_code 0
06:46:46 (in ustcomm_register_enum() at lttng-ust-comm.c:1622)
06:46:46 double free or corruption (!prev)
06:46:46 Aborted
```

The sessiond log reports an error serialising the metadata ([Full log](#)) :

```
DEBUG3 - 10:46:46.312634981 [1265363/1265375]: Append to metadata: "floating_point { exp_dig = 8;
mant_dig = 24; align = 8; } _float;
" (in lttng_metadata_printf() at ust-metadata.c:166)
Error: Error appending channel metadata (errno = -75)
```

...

Error: 1 trace chunks are leaked by lttng-consumerd. This can be caused by an internal error of the session daemon.

Error: consumer closed the command socket

Error: Health error occurred in thread_consumer_management

Running gdb on the jvm during the test shows the double free happening in 'ustcomm_register_channel' on 'fields'

```
0 0x00007f1dd6b68ce5 in raise () from /usr/lib/libc.so.6
No symbol table info available.
#1 0x00007f1dd6b52857 in abort () from /usr/lib/libc.so.6
No symbol table info available.
#2 0x00007f1dd6bac2b0 in __libc_message () from /usr/lib/libc.so.6
No symbol table info available.
#3 0x00007f1dd6bb374a in malloc_printerr () from /usr/lib/libc.so.6
No symbol table info available.
#4 0x00007f1dd6bb55cc in _int_free () from /usr/lib/libc.so.6
No symbol table info available.
#5 0x00007f1da04d94e7 in ustcomm_register_channel (sock=15, session=0x7f1d5c000c70, session_objd=
1, channel_objd=2, nr_ctx_fields=1, ctx_fields=0x7f1d5c013f80, chan_id=0x7f1d997fcd8, header_type
=0x7f1d5c013db8) at lttng-ust-comm.c:1697
    len = 11960
    msg = {header = {notify_cmd = 1}, m = {session_objd = 1, channel_objd = 2, ctx_fields_len
= 11960, padding = '\000' <repeats 31 times>}}
    reply = {header = {notify_cmd = 1818388844}, r = {ret_code = 1735292020, chan_id = 1953723
743, header_type = 1634890847, padding = "cepoint[185588/185609]: Release "}}
    fields_len = 11960
    fields = 0x7f1d5c01c2e0
    ret = 0
    nr_write_fields = 13
    __func__ = "ustcomm_register_channel"
#6 0x00007f1da04ef03c in lttng_session_enable (session=0x7f1d5c000c70) at lttng-events.c:458
    ctx = 0x7f1d5c013f50
    fields = 0x7f1d5c013f80
    nr_fields = 1
    chan_id = 4294967295
    ret = 0
    chan = 0x7f1d5c013d70
    notify_socket = 15
    __func__ = "lttng_session_enable"
#7 0x00007f1da04e46d7 in lttng_session_cmd (objd=1, cmd=128, arg=139765106071992, uargs=0x7f1d997
fd290, owner=0x7f1da0575c00 <local_apps>) at lttng-ust-abi.c:580
    session = 0x7f1d5c000c70
```

To reproduce the problem, build the latest master from ust and tools with the java agents enabled and debug symbols. Checkout the [lttng-ust-java-test](#) repo and use the following script to run the test, the only additional requirement on top of the build dependencies is 'maven' which you should be packaged in most distro :

```
#!/bin/sh
```

```

PREFIX="$HOME/opt"

LOG4J_JAR="$PREFIX/share/java/log4j-1.2.17.jar"

# To override the default java jvm
#export JAVA_HOME=/usr/lib/jvm/${java_version}-${arch}

export LD_LIBRARY_PATH=${PREFIX}/lib
export PATH=${PREFIX}/bin:${PATH}
export LTTNG_UST_DEBUG=1
export LTTNG_CONSUMERD32_BIN=${PREFIX}/lib/lttng/libexec/lttng-consumerd
export LTTNG_CONSUMERD64_BIN=${PREFIX}/lib/lttng/libexec/lttng-consumerd
export LTTNG_SESSION_CONFIG_XSD_PATH=${PREFIX}/share/xml/lttng

# Start the lttng-sessiond
lttng-sessiond -b -vvv >lttng-sessiond.log 2>&1

mvn \
  -Dmaven.test.failure.ignore=true \
  -DfailIfNoTests=false \
  -Dlog4j-jar-location=${LOG4J_JAR} \
  -Dcommon-jar-location=${PREFIX}/share/java/lttng-ust-agent-common.jar \
  -Djul-jar-location=${PREFIX}/share/java/lttng-ust-agent-jul.jar \
  -Dlog4j-jar-location=${PREFIX}/share/java/lttng-ust-agent-log4j.jar \
  -DargLine=-Djava.library.path=${PREFIX}/lib \
  -Dtest='JulAppContextOrderingIT#testSessionAgentRetriever' \
  clean verify

# Kill the sessiond
killall lttng-sessiond

```

The only tricky part is attaching gdb to the correct jvm, I haven't yet found a reliable way to fully automate this step. You need to run the previously mentioned script and wait for the final phase of the test to start before starting gdb. You have to wait for the second time you see coloured output before attaching gdb.

Attach gdb to the jvm running the test (disable signals used internally by the jvm)

```

gdb --init-eval-command='handle SIGSEGV nostop noprint pass' --init-eval-command='handle SIG62 nos
top noprint pass' -p $(jps | grep -i sure | awk '{ print $1 }')

```

History

#1 - 04/22/2020 01:21 PM - Mathieu Desnoyers

- Status changed from New to Resolved

- Project changed from LTTng to LTTng-UST

Fixed by:

```

commit cf22367fec17f2b2650e637de84fc2f614e4d2ac
Author: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
Date: Wed Apr 22 12:22:19 2020 -0400

```

Fix: incorrect field accounting for dynamic type

Two field accounting issues were introduced by commit 218deb69baab "tracepoint: Refactor representation of nested types" which affect the dynamic types used by Java application contexts.

The structure types should account 1 for the structure per se, in addition to the nested fields.

The atype_dynamic includes an implicit enumeration, which now has two fields instead of 1.

This was causing memory corruption issues in the lttng-ust-java-tests.

Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>