LTTng-modules - Bug #1167

Linux Kernel Oops when creating LTTng kernel channel while using huge pages

05/31/2018 09:34 AM - Aleix Roca Nonell

Status:	Resolved	**Start date:**	05/31/2018
Priority:	Normal	**Due date:**	
Assignee:	Mathieu Desnoyers	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

I have stumbled with a Linux Kernel Oops while using LTTng kernel tracer. The bug pops up every time I try to trace a simple application that makes use of 1G hugepages with libhugepagetlbfs. Please note that if the LTTng channel buffer size is not big enough (I have not determined exactly what enough is) the Oops does not appears.

To bug should be reproducible with the following commands:

lttng enable-channel --kernel --num-subbuf=16 --subbuf-size=128M big-krn-channel
lttng enable-event --kernel --channel=big-krn-channel --all
lttng start
hugectl --heap=1G sleep 10
lttng stop
lttng destroy

I'm using the latest lttng 2.10 ust(2.10.1)/tools(2.10.4)/modules(2.10.6) from git.

LTTng-ust: 979c384814c9fad78c1e297a129da86dda2e4573
LTTng-modules: 4c517fb46949a6dfcaf73fa2131b730432ac640b
LTTng-tools: a0b75f6cfee839d0f85fab6bb2c0d8e80db5823d

The system has two sockets with a total of 56 CPUs and 128GiB of main memory.
ThI'm using a linux kernel 4.15.9 (and I have seen the same problem on a linux 4.15.18).

The Oops message follows:

xeon08 login: [244.741758] BUG: unable to handle kernel NULL pointer dereference at 0000000000000000
[244.750940] IP: *_cpuhp_state_remove_instance+0xc1/0x1c0*
[244.757246] PGD 80000020390b2067 P4D 80000020390b2067 PUD 2036ad9067 PMD 0
[244.765445] Oops: 0002 [#1] SMP PTI
[244.769713] Modules linked in: msr(E) lttng_probe_x86_exceptions(OE) lttng_probe_x86_irq_vectors(OE) lttng_probe_writeback(OE) lttng_probe_workqueue(OE) lttng_probe_vmscan(OE) lttng_probe_udp(OE) lttng_probe_timer(OE) lttng_probe_sunrpc(OE) lttng_probe_statedump(OE) lttng_probe_sock(OE) lttng_probe_skb(OE) lttng_probe_signal(OE) lttng_probe_scsi(OE) lttng_probe_sched(OE) lttng_probe_regmap(OE) lttng_probe_rcu(OE) lttng_probe_random(OE) lttng_probe_printk(OE) lttng_probe_power(OE) lttng_probe_net(OE) lttng_probe_napi(OE) lttng_probe_module(OE) lttng_probe_kvm_x86_mmu(OE) lttng_probe_kvm_x86(OE) kvm(E) irqbypass(E) lttng_probe_kvm(OE) lttng_probe_kmem(OE) lttng_probe_jbd2(OE) lttng_probe_irq(OE) lttng_probe_i2c(OE) lttng_probe_gpio(OE) lttng_probe_ext4(OE) lttng_probe_compaction(OE) lttng_probe_block(OE)
[244.852692] lttng_ring_buffer_metadata_mmap_client(OE) lttng_ring_buffer_client_mmap_overwrite(OE) lttng_ring_buffer_client_mmap_discard(OE) lttng_ring_buffer_metadata_client(OE) lttng_ring_buffer_client_overwrite(OE) lttng_ring_buffer_client_discard(OE) lttng_tracer(OE) lttng_statedump(OE) lttng_ftrace(OE) lttng_kprobes(OE) lttng_clock(OE) lttng_lib_ring_buffer(OE) lttng_kretprobes(OE) nfsv3(E) nfs_acl(E) nfs(E) lockd(E) grace(E) sunrpc(E) fscache(E) af_packet(E) iscsi_ibft(E) iscsi_boot_sysfs(E) intel_rapl(E) sb_edac(E) x86_pkg_temp_thermal(E) intel_powerclamp(E) mgag200(E) hfi1(E) coretemp(E) drm_kms_helper(E) crct10dif_pclmul(E) syscopyarea(E) crc32_pclmul(E) sysfillrect(E) crc32c_intel(E) sysimgblt(E) fb_sys_fops(E) ttm(E) ixgbe(E) aesni_intel(E) drm(E) rdmavt(E) aes_x86_64(E) mdio(E) iTCO_wdt(E)
[244.852724] ipmi_si(E) joydev(E) ptp(E) crypto_simd(E) iTCO_vendor_support(E) i2c_algo_bit(E) ipmi_devintf(E) cryptd(E) pps_core(E) glue_helper(E) mei_me(E) ipmi_msghandler(E) ioatdma(E) pcspkr(E) lpc_ich(E) mei(E) mfd_core(E) i2c_i801(E) dca(E) shpchp(E) wmi(E) acpi_pad(E) ib_ipoib(E) acpi_cpufreq(E) rdma_ucm(E) button(E) ib_ucm(E) ib_uverbs(E) ib_umad(E) rdma_cm(E) configfs(E) ib_cm(E) iw_cm(E) ib_core(E) ext4(E) crc16(E) mbcache(E) jbd2(E) hid_generic(E) usbhid(E) sd_mod(E) xhci_pci(E) xhci_hcd(E) ehci_pci(E) ehci_hcd(E) ahci(E) libahci(E) libata(E) usbcore(E) sg(E) scsi_mod(E) autofs4(E)
[244.852757] CPU: 36 PID: 2401 Comm: lttng-sessiond Tainted: G OE 4.15.9-stable #4
[244.852758] Hardware name: Intel Corporation S2600WTTR/S2600WTTR, BIOS SE5C610.86B.01.01.0016.033120161139 03/31/2016
*[244.852760] RIP: 0010:*_cpuhp_state_remove_instance+0xc1/0x1c0

```
[  244.852761] RSP: 0018:ffffc9000996fa98 EFLAGS: 00010246
[  244.852762] RAX: 0000000000000000 RBX: 0000000000000038 RCX: 0000000000000000
[  244.852763] RDX: 0000000000000000 RSI: 0000000000000038 RDI: ffffffff822fbea0
[  244.852763] RBP: 0000000000000041 R08: 0000000000000000 R09: 0000000000000000
[  244.852764] R10: 0000000000000000 R11: 0000000000000001 R12: ffff88202d29bc68
[  244.852764] R13: 0000000000016aa0 R14: 0000000000000a28 R15: 0000000000000000
[  244.852765] FS:  00007fc00effd700(0000) GS:ffff88103f200000(0000) knlGS:0000000000000000
[  244.852766] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  244.852767] CR2: 0000000000000000 CR3: 0000002038570002 CR4: 00000000003606e0
[  244.852768] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[  244.852768] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
[  244.852769] Call Trace:
[  244.852776]  ? channel_backend_init+0x1dc/0x2f0 [lttng_lib_ring_buffer]
[  244.852780]  channel_backend_init+0x1dc/0x2f0 [lttng_lib_ring_buffer]
[  244.852783]  channel_create+0x6d/0x1b0 [lttng_lib_ring_buffer]
[  244.852788]  _channel_create+0x34/0x80 [lttng_ring_buffer_client_discard]
[  244.852831]  lttng_channel_create+0xff/0x1c0 [lttng_tracer]
[  244.852843]  lttng_abi_create_channel+0x119/0x220 [lttng_tracer]
[  244.852855]  lttng_session_ioctl+0x19e/0x2e0 [lttng_tracer]
[  244.852859]  ? hrtimer_try_to_cancel+0xc4/0x110
[  244.852862]  ? hrtimer_try_to_cancel+0xc4/0x110
[  244.852864]  ? hrtimer_cancel+0x15/0x20
[  244.852867]  ? futex_wait+0x177/0x220
[  244.852872]  ? sched_clock+0x5/0x10
[  244.852875]  ? sched_clock_cpu+0xc/0xa0
[  244.852877]  ? __lock_acquire.isra.31+0x165/0x700
[  244.852886]  do_vfs_ioctl+0x8f/0x5e0
[  244.852890]  ? __fget+0xb4/0xf0
[  244.852891]  ? __fget+0x5/0xf0
[  244.852893]  SyS_ioctl+0x74/0x80
[  244.852898]  do_syscall_64+0x6e/0x1a0
[  244.852901]  entry_SYSCALL_64_after_hwframe+0x3d/0xa2
[  244.852902] RIP: 0033:0x7fc0154254b7
[  244.852903] RSP: 002b:00007fc00efec5b8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010
[  244.852904] RAX: ffffffffffffffda RBX: 00007fc000005410 RCX: 00007fc0154254b7
[  244.852905] RDX: 00007fc00efec5c0 RSI: 000000004140f655 RDI: 0000000000000030
[  244.852905] RBP: 00007fc000005470 R08: 0000000000000000 R09: 0000000000000030
[  244.852906] R10: ffffffffffffffe80 R11: 0000000000000246 R12: 00000000ffffffff
[  244.852907] R13: 00007fc000001b70 R14: 0000000000000050 R15: 00007fc000001b70
[  244.852910] Code: 15 25 b8 fd 00 85 d2 0f 85 f9 00 00 00 31 f6 48 c7 c7 00 b8 05 82 e8 8f c0 63 00 84 db 75 75 49 8b 04 24 49
8b 54 24 08 48 85 c0 <48> 89 02 74 04 48 89 50 08 48 b8 00 01 00 00 00 00 ad de 48 c7
[  244.852934] RIP: __cpuhp_state_remove_instance+0xc1/0x1c0 RSP: ffffc9000996fa98
[  244.852935] CR2: 0000000000000000
[  244.852937] ---[ end trace a8be70506d9817e6 ]---
```

Associated revisions

Revision fbbb0005 - 09/07/2018 05:56 PM - Mathieu Desnoyers

Fix: out of memory error handling

CPU hotplug handles teardown on failure to complete adding an instance
of CPU hotplug. Trying to remove after a failed "add" on that instance
triggers a NULL pointer dereference OOPS.

Fixes: #1167
Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>

Revision 5f14d8ae - 09/07/2018 05:58 PM - Mathieu Desnoyers

Fix: out of memory error handling

CPU hotplug handles teardown on failure to complete adding an instance
of CPU hotplug. Trying to remove after a failed "add" on that instance
triggers a NULL pointer dereference OOPS.

Fixes: #1167
Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>

Revision 9837fedc - 09/07/2018 05:58 PM - Mathieu Desnoyers

Fix: out of memory error handling

CPU hotplug handles teardown on failure to complete adding an instance
of CPU hotplug. Trying to remove after a failed "add" on that instance
triggers a NULL pointer dereference OOPS.

Fixes: #1167
Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>

Revision 33a72020 - 09/07/2018 05:58 PM - Mathieu Desnoyers

Fix: out of memory error handling

CPU hotplug handles teardown on failure to complete adding an instance
of CPU hotplug. Trying to remove after a failed "add" on that instance
triggers a NULL pointer dereference OOPS.

Fixes: #1167
Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>

History

#1 - 09/07/2018 06:00 PM - Mathieu Desnoyers

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset lttng-modules|5f14d8ae2cc0734b007c8770c3b13ff00d830040.

#2 - 09/07/2018 06:01 PM - Mathieu Desnoyers

- Assignee set to Mathieu Desnoyers

- % Done changed from 100 to 0

The situation is triggered by an out-of-memory due to trying to allocate too large buffers for the physical memory of the machine.

The NULL pointer OOPS is not expected though. I pushed a fix for that issue upstream:

```
commit 5f14d8ae2cc0734b007c8770c3b13ff00d830040
Author: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
Date:   Fri Sep 7 17:55:32 2018 -0400

    Fix: out of memory error handling

    CPU hotplug handles teardown on failure to complete adding an instance
    of CPU hotplug. Trying to remove after a failed "add" on that instance
    triggers a NULL pointer dereference OOPS.

    Fixes: #1167
    Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
```

The problem shows up at enable-channel command, so it seems unrelated to huge pages.