



author Stefan Assmann <sassmann@kpanic.de> 2023-03-13 17:06:45 +0100
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-30 12:47:47 +0200
commit [7a29799fc141ba9e6cf921fc8e958e3398ad1a4f](#) (patch)
tree [451b36e4b63832cbecbfc2903701705a5eea21f7](#)
parent [ce19c70f308fe3360e55cd51eaacd376c599fe54](#) (diff)
download [linux-7a29799fc141ba9e6cf921fc8e958e3398ad1a4f.tar.gz](#)

diff options

context:
space:
mode:

iavf: fix hang on reboot with ice

[Upstream commit 4e264be98b88a6d6f476c11087fe865696e8bef5]

When a system with E810 with existing VFs gets rebooted the following hang may be observed.

```
Pid 1 is hung in iavf_remove(), part of a network driver:
PID: 1      TASK: ffff965400e5a340  CPU: 24   COMMAND: "systemd-shutdown"
#0 [ffffaad04005fa50] __schedule at ffffffff8b3239cb
#1 [ffffaad04005fae8] schedule at ffffffff8b323e2d
#2 [ffffaad04005fb00] schedule_hrtimeout_range_clock at ffffffff8b32cebc
#3 [ffffaad04005fb80] usleep_range_state at ffffffff8b32c930
#4 [ffffaad04005fbb0] iavf_remove at ffffffff8b32b4c [iavf]
#5 [ffffaad04005fbb0] pci_device_remove at ffffffff8b32b4c [iavf]
#6 [ffffaad04005fc10] device_release_driver_internal at ffffffff8b32b4c [iavf]
#7 [ffffaad04005fc40] pci_stop_bus_device at ffffffff8b32b4c [iavf]
#8 [ffffaad04005fc60] pci_stop_and_remove_bus_device at ffffffff8b32b4c [iavf]
#9 [ffffaad04005fc70] pci_iov_remove_virtfn at ffffffff8b32b4c [iavf]
#10 [ffffaad04005fca8] sriov_disable at ffffffff8b32b4c [iavf]
#11 [ffffaad04005fcc8] ice_free_vfs at ffffffff8b32b4c [ice]
#12 [ffffaad04005fd10] ice_remove at ffffffff8b32b4c [ice]
#13 [ffffaad04005fd38] ice_shutdown at ffffffff8b32b4c [ice]
#14 [ffffaad04005fd50] pci_device_shutdown at ffffffff8b32b4c [iavf]
#15 [ffffaad04005fd70] device_shutdown at ffffffff8b32b4c [iavf]
#16 [ffffaad04005fd98] kernel_restart at ffffffff8b32b4c [iavf]
#17 [ffffaad04005fda8] __do_sys_reboot at ffffffff8b32b4c [iavf]
#18 [ffffaad04005fee0] do_syscall_64 at ffffffff8b32b4c [iavf]
#19 [ffffaad04005ff08] __context_tracking_enter at ffffffff8b32b4c [iavf]
#20 [ffffaad04005ff18] syscall_exit_to_user_mode at ffffffff8b32b4c [iavf]
#21 [ffffaad04005ff28] do_syscall_64 at ffffffff8b32b4c [iavf]
#22 [ffffaad04005ff50] entry_SYSCALL_64_after_hwframe at ffffffff8b32b4c [iavf]
RIP: 00007f1baa5c13d7  RSP: 00007fffbcc55a98  RFLAGS: 00000202
RAX: ffffffff8b32b4c  RBX: 0000000000000000  RCX: 00007f1baa5c13d7
RDX: 0000000000123456  RSI: 000000000028121969  RDI: 0000000000fee1dead
RBP: 00007fffbcc55ca0  R8: 0000000000000000  R9: 00007fffbcc54e90
R10: 00007fffbcc55050  R11: 00000000000000202  R12: 0000000000000005
R13: 0000000000000000  R14: 00007fffbcc55af0  R15: 0000000000000000
ORIG_RAX: 00000000000000a9  CS: 0033  SS: 002b
```

During reboot all drivers PM shutdown callbacks are invoked.
In iavf_shutdown() the adapter state is changed to __IAVF_REMOVE.
In ice_shutdown() the call chain above is executed, which at some point calls iavf_remove(). However iavf_remove() expects the VF to be in one of the states __IAVF_RUNNING, __IAVF_DOWN or __IAVF_INIT_FAILED. If that's not the case it sleeps forever.
So if iavf_shutdown() gets invoked before iavf_remove() the system will

hang indefinitely because the adapter is already in state __IAVF_REMOVE.

Fix this by returning from iavf_remove() if the state is __IAVF_REMOVE, as we already went through iavf_shutdown().

Fixes: 974578017fc1 ("iavf: Add waiting so the port is initialized in remove")

Fixes: a8417330f8a5 ("iavf: Fix race condition between iavf_shutdown and iavf_remove")

Reported-by: Marius Cornea <mcornea@redhat.com>

Signed-off-by: Stefan Assmann <sassmann@kpanic.de>

Reviewed-by: Michal Kubiak <michal.kubiak@intel.com>

Tested-by: Rafal Romanowski <rafal.romanowski@intel.com>

Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>

Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r-- drivers/net/ethernet/intel/iavf/iavf_main.c 5
--

1 files changed, 5 insertions, 0 deletions

diff --git a/drivers/net/ethernet/intel/iavf/iavf_main.c b/drivers/net/ethernet/intel/iavf/iavf_main.c

index 82c4f1190e41c8..f5e6ae2c683f4e 100644

--- a/drivers/net/ethernet/intel/iavf/iavf_main.c

+++ b/drivers/net/ethernet/intel/iavf/iavf_main.c

```
@@ -4213,6 +4213,11 @@ static void iavf_remove(struct pci_dev *pdev)
        mutex_unlock(&adapter->crit_lock);
        break;
    }
+   /* Simply return if we already went through iavf_shutdown */
+   if (adapter->state == __IAVF_REMOVE) {
+       mutex_unlock(&adapter->crit_lock);
+       return;
+   }

    mutex_unlock(&adapter->crit_lock);
    usleep_range(500, 1000);
```