



author Stefan Assmann <sassmann@kpanic.de> 2023-03-13 17:06:45 +0100
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-30 12:49:07 +0200
commit [502b898235f06130750c91512c86dd0e9efe28e6](#) (patch)
tree [7ae5f100ab4cf27cabe42e41a53431958acf6e49](#)
parent [e16fa0c9ff7c59f970a426f9f999ef017d884ec9](#) (diff)
download [linux-502b898235f06130750c91512c86dd0e9efe28e6.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 [fffffaad04005fa50] __schedule at ffffffff8b3239cb
#1 [fffffaad04005fae8] schedule at ffffffff8b323e2d
#2 [fffffaad04005fb00] schedule_hrtimeout_range_clock at ffffffff8b32cebc
#3 [fffffaad04005fb80] usleep_range_state at ffffffff8b32c930
#4 [fffffaad04005fbb0] iavf_remove at ffffffff8b32b4c [iavf]
#5 [fffffaad04005fbbf] pci_device_remove at ffffffff8b32b7513
#6 [fffffaad04005fcb0] device_release_driver_internal at ffffffff8b32b8aa
#7 [fffffaad04005fcbf] pci_stop_bus_device at ffffffff8b32b95fc
#8 [fffffaad04005fcd0] pci_stop_and_remove_bus_device at ffffffff8b32ba81e
#9 [fffffaad04005fcd7] pci_iov_remove_virtfn at ffffffff8b32ba929
#10 [fffffaad04005fcd8] sriov_disable at ffffffff8b32ba9e4
#11 [fffffaad04005fcd9] ice_free_vfs at ffffffff8b32bb2c8 [ice]
#12 [fffffaad04005fcd0] ice_remove at ffffffff8b32bb7fe [ice]
#13 [fffffaad04005fcd38] ice_shutdown at ffffffff8b32bb7946 [ice]
#14 [fffffaad04005fcd50] pci_device_shutdown at ffffffff8b32ba58f1
#15 [fffffaad04005fcd70] device_shutdown at ffffffff8b32ba05386
#16 [fffffaad04005fcd98] kernel_restart at ffffffff8b32ba92a870
#17 [fffffaad04005fda8] __do_sys_reboot at ffffffff8b32abd6
#18 [fffffaad04005fee0] do_syscall_64 at ffffffff8b32b17159
#19 [fffffaad04005ff08] __context_tracking_enter at ffffffff8b32b1b6fc
#20 [fffffaad04005ff18] syscall_exit_to_user_mode at ffffffff8b32b1b50d
#21 [fffffaad04005ff28] do_syscall_64 at ffffffff8b32b17169
#22 [fffffaad04005ff50] entry_SYSCALL_64_after_hwframe at ffffffff8b32b009b
RIP: 00007f1baa5c13d7  RSP: 00007fffbcc55a98  RFLAGS: 00000202
RAX: ffffffff8b32b009b  RBX: 0000000000000000  RCX: 00007f1baa5c13d7
RDX: 00000000001234567  RSI: 000000000028121969  RDI: 0000000000fee1dead
RBP: 00007fffbcc55ca0   R8: 0000000000000000   R9: 00007fffbcc554e90
R10: 00007fffbcc55050  R11: 00000000000000202  R12: 00000000000000005
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 41edcf9e154a3b..5f8fff6c701fc9 100644

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

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

```
@@ -5086,6 +5086,11 @@ 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);
```