author	Dean Luick <dean.luick@cornelisnetworks.com>	2022-10-18 10:27:50 -0400
committer	Leon Romanovsky <leon@kernel.org>	2022-10-19 10:06:07 +0300
commit	1afac08b39d85437187bb2a92d89a741b1078f55 (patch)	
tree	cd0817bb40ff85d1665d1acf0045f83a44c6baeb	
parent	eb83f502adb036cd56c27e13b9ca3b2aabfa790b (diff)	
download	linux-1afac08b39d85437187bb2a92d89a741b1078f55.tar.gz	

diff options

context: 3

space: include

mode: unified

IB/hfi1: Correctly move list in sc_disable()

```
Commit 13bac861952a ("IB/hfi1: Fix abba locking issue with sc_disable()")
incorrectly tries to move a list from one list head to another.  The
result is a kernel crash.

The crash is triggered when a link goes down and there are waiters for a
send to complete.  The following signature is seen:

  BUG: kernel NULL pointer dereference, address: 0000000000000030
  [...]
  Call Trace:
   sc_disable+0x1ba/0x240 [hfi1]
   pio_freeze+0x3d/0x60 [hfi1]
   handle_freeze+0x27/0x1b0 [hfi1]
   process_one_work+0x1b0/0x380
   ? process_one_work+0x380/0x380
   worker_thread+0x30/0x360
   ? process_one_work+0x380/0x380
   kthread+0xd7/0x100
   ? kthread_complete_and_exit+0x20/0x20
   ret_from_fork+0x1f/0x30

The fix is to use the correct call to move the list.

Fixes: 13bac861952a ("IB/hfi1: Fix abba locking issue with sc_disable()")
Signed-off-by: Dean Luick <dean.luick@cornelisnetworks.com>
Signed-off-by: Dennis Dalessandro <dennis.dalessandro@cornelisnetworks.com>
Link: https://lore.kernel.org/r/166610327042.674422.6146908799669288976.stgit@awfm-02.cornelisnetworks.com
Signed-off-by: Leon Romanovsky <leon@kernel.org>
```

Diffstat

-rw-r--r-- drivers/infiniband/hw/hfi1/pio.c 3

1 files changed, 1 insertions, 2 deletions

```
diff --git a/drivers/infiniband/hw/hfi1/pio.c b/drivers/infiniband/hw/hfi1/pio.c
index 3d42bd2b36bd43..51ae58c02b15c7 100644
--- a/drivers/infiniband/hw/hfi1/pio.c
+++ b/drivers/infiniband/hw/hfi1/pio.c
@@ -913,8 +913,7 @@ void sc_disable(struct send_context *sc)
        spin_unlock(&sc->release_lock);

        write_seqlock(&sc->waitlock);
-       if (!list_empty(&sc->piowait))
-               list_move(&sc->piowait, &wake_list);
+       list_splice_init(&sc->piowait, &wake_list);
        write_sequnlock(&sc->waitlock);
        while (!list_empty(&wake_list)) {
```

```
struct iowait *wait;
```