author	Shannon Nelson <shannon.nelson@amd.com>	2025-04-21 10:46:06 -0700
committer	Jakub Kicinski <kuba@kernel.org>	2025-04-23 18:50:17 -0700
commit	3f77c3dfffc7063428b100c4945ca2a7a8680380 (patch)	
tree	09ecbbac33d94497146f8a3265ddfd00c9557637	
parent	f9559d818205a4a0b9cd87181ef46e101ea11157 (diff)	
download	linux-3f77c3dfffc7063428b100c4945ca2a7a8680380.tar.gz	

diff options

context: 3

space: include

mode: unified

pds_core: make wait_context part of q_info

```
Make the wait_context a full part of the q_info struct rather
than a stack variable that goes away after pdsc_adminq_post()
is done so that the context is still available after the wait
loop has given up.

There was a case where a slow development firmware caused
the adminq request to time out, but then later the FW finally
finished the request and sent the interrupt.  The handler tried
to complete_all() the completion context that had been created
on the stack in pdsc_adminq_post() but no longer existed.
This caused bad pointer usage, kernel crashes, and much wailing
and gnashing of teeth.

Fixes: 01ba61b55b20 ("pds_core: Add adminq processing and commands")
Reviewed-by: Simon Horman <horms@kernel.org>
Signed-off-by: Shannon Nelson <shannon.nelson@amd.com>
Reviewed-by: Jacob Keller <jacob.e.keller@intel.com>
Link: https://patch.msgid.link/20250421174606.3892-5-shannon.nelson@amd.com
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
```

Diffstat

-rw-r--r--	drivers/net/ethernet/amd/pds_core/adminq.c	36
-rw-r--r--	drivers/net/ethernet/amd/pds_core/core.c	4
-rw-r--r--	drivers/net/ethernet/amd/pds_core/core.h	2

3 files changed, 18 insertions, 24 deletions

```
diff --git a/drivers/net/ethernet/amd/pds_core/adminq.c b/drivers/net/ethernet/amd/pds_core/adminq.c
index c83a0a80d5334e..506f682d15c10a 100644
--- a/drivers/net/ethernet/amd/pds_core/adminq.c
+++ b/drivers/net/ethernet/amd/pds_core/adminq.c
@@ -5,11 +5,6 @@

 #include "core.h"

-struct pdsc_wait_context {
-       struct pdsc_qcq *qcq;
-       struct completion wait_completion;
-};
-
 static int pdsc_process_notifyq(struct pdsc_qcq *qcq)
 {
        union pds_core_notifyq_comp *comp;
@@ -109,10 +104,10 @@ void pdsc_process_adminq(struct pdsc_qcq *qcq)
```

```
                q_info = &q->info[q->tail_idx];
                q->tail_idx = (q->tail_idx + 1) & (q->num_descs - 1);

-               /* Copy out the completion data */
-               memcpy(q_info->dest, comp, sizeof(*comp));
-
-               complete_all(&q_info->wc->wait_completion);
+               if (!completion_done(&q_info->completion)) {
+                       memcpy(q_info->dest, comp, sizeof(*comp));
+                       complete(&q_info->completion);
+               }

                if (cq->tail_idx == cq->num_descs - 1)
                        cq->done_color = !cq->done_color;
@@ -162,8 +157,7 @@ irqreturn_t pdsc_adminq_isr(int irq, void *data)
 static int __pdsc_adminq_post(struct pdsc *pdsc,
                              struct pdsc_qcq *qcq,
                              union pds_core_adminq_cmd *cmd,
-                             union pds_core_adminq_comp *comp,
-                             struct pdsc_wait_context *wc)
+                             union pds_core_adminq_comp *comp)
 {
        struct pdsc_queue *q = &qcq->q;
        struct pdsc_q_info *q_info;
@@ -205,9 +199,9 @@ static int __pdsc_adminq_post(struct pdsc *pdsc,
        /* Post the request */
        index = q->head_idx;
        q_info = &q->info[index];
-       q_info->wc = wc;
        q_info->dest = comp;
        memcpy(q_info->desc, cmd, sizeof(*cmd));
+       reinit_completion(&q_info->completion);

        dev_dbg(pdsc->dev, "head_idx %d tail_idx %d\n",
                q->head_idx, q->tail_idx);
@@ -231,16 +225,13 @@ int pdsc_adminq_post(struct pdsc *pdsc,
                     union pds_core_adminq_comp *comp,
                     bool fast_poll)
 {
-       struct pdsc_wait_context wc = {
-               .wait_completion =
-                       COMPLETION_INITIALIZER_ONSTACK(wc.wait_completion),
-       };
        unsigned long poll_interval = 1;
        unsigned long poll_jiffies;
        unsigned long time_limit;
        unsigned long time_start;
        unsigned long time_done;
        unsigned long remaining;
+       struct completion *wc;
        int err = 0;
        int index;

@@ -250,20 +241,19 @@ int pdsc_adminq_post(struct pdsc *pdsc,
                return -ENXIO;
        }

-       wc.qcq = &pdsc->adminqcq;
-       index = __pdsc_adminq_post(pdsc, &pdsc->adminqcq, cmd, comp, &wc);
+       index = __pdsc_adminq_post(pdsc, &pdsc->adminqcq, cmd, comp);
        if (index < 0) {
                err = index;
                goto err_out;
        }
```

```
+       wc = &pdsc->adminqcq.q.info[index].completion;
        time_start = jiffies;
        time_limit = time_start + HZ * pdsc->devcmd_timeout;
        do {
                /* Timeslice the actual wait to catch IO errors etc early */
                poll_jiffies = msecs_to_jiffies(poll_interval);
-               remaining = wait_for_completion_timeout(&wc.wait_completion,
-                                                       poll_jiffies);
+               remaining = wait_for_completion_timeout(wc, poll_jiffies);
                if (remaining)
                        break;

@@ -292,9 +282,11 @@ int pdsc_adminq_post(struct pdsc *pdsc,
        dev_dbg(pdsc->dev, "%s: elapsed %d msecs\n",
                __func__, jiffies_to_msecs(time_done - time_start));

-       /* Check the results */
-       if (time_after_eq(time_done, time_limit))
+       /* Check the results and clear an un-completed timeout */
+       if (time_after_eq(time_done, time_limit) && !completion_done(wc)) {
                err = -ETIMEDOUT;
+               complete(wc);
+       }

        dev_dbg(pdsc->dev, "read admin queue completion idx %d:\n", index);
        dynamic_hex_dump("comp ", DUMP_PREFIX_OFFSET, 16, 1,
```

diff --git a/drivers/net/ethernet/amd/pds_core/core.c b/drivers/net/ethernet/amd/pds_core/core.c
index 55163457f12bec..9512aa4083f054 100644
--- a/drivers/net/ethernet/amd/pds_core/core.c
+++ b/drivers/net/ethernet/amd/pds_core/core.c

```
@@ -167,8 +167,10 @@ static void pdsc_q_map(struct pdsc_queue *q, void *base, dma_addr_t base_pa)
        q->base = base;
        q->base_pa = base_pa;

-       for (i = 0, cur = q->info; i < q->num_descs; i++, cur++)
+       for (i = 0, cur = q->info; i < q->num_descs; i++, cur++) {
                cur->desc = base + (i * q->desc_size);
+               init_completion(&cur->completion);
+       }
 }

 static void pdsc_cq_map(struct pdsc_cq *cq, void *base, dma_addr_t base_pa)
```

diff --git a/drivers/net/ethernet/amd/pds_core/core.h b/drivers/net/ethernet/amd/pds_core/core.h
index 199473112c295c..0b53a1fab46d02 100644
--- a/drivers/net/ethernet/amd/pds_core/core.h
+++ b/drivers/net/ethernet/amd/pds_core/core.h

```
@@ -96,7 +96,7 @@ struct pdsc_q_info {
        unsigned int bytes;
        unsigned int nbufs;
        struct pdsc_buf_info bufs[PDS_CORE_MAX_FRAGS];
-       struct pdsc_wait_context *wc;
+       struct completion completion;
        void *dest;
 };
```