



author Robin Murphy <robin.murphy@arm.com> 2025-04-10 12:23:48 +0100
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2025-05-02 07:59:25 +0200
commit [b14d98641312d972bb3f38e82eddf92898522389](#) (patch)
tree [793acb39aad7b927abdf657604b648d6ecc14054](#)
parent [71c3d43c8f7046eb8b5a29f0c4f51a005b7a0f8f](#) (diff)
download [linux-b14d98641312d972bb3f38e82eddf92898522389.tar.gz](#)

diff options

context: 3 ▼
space: include ▼
mode: unified ▼

iommu: Clear iommu-dma ops on cleanup

[Upstream commit 280e5a30100578106a4305ce0118e0aa9b866f12]

If `iommu_device_register()` encounters an error, it can end up tearing down already-configured groups and default domains, however this currently still leaves devices hooked up to iommu-dma (and even historically the behaviour in this area was at best inconsistent across architectures/drivers...) Although in the case that an IOMMU is present whose driver has failed to probe, users cannot necessarily expect DMA to work anyway, it's still arguable that we should do our best to put things back as if the IOMMU driver was never there at all, and certainly the potential for crashing in iommu-dma itself is undesirable. Make sure we clean up the `dev->dma_iommu` flag along with everything else.

Reported-by: Chen-Yu Tsai <wenst@chromium.org>

Signed-off-by: Robin Murphy <robin.murphy@arm.com>

Closes: <https://lore.kernel.org/all/CAGXv+5HJpTYmQ2h-GD7GjyeYT7bL9EBCvu0mz5LgpzJZtzfW0w@mail.gmail.com/>

Tested-by: Chen-Yu Tsai <wenst@chromium.org>

Reviewed-by: Lu Baolu <baolu.lu@linux.intel.com>

Link: <https://lore.kernel.org/r/e788aa927f6d827dd4ea1ed608fada79f2bab030.1744284228.git.robin.murphy@arm.com>

Signed-off-by: Joerg Roedel <jroedel@suse.de>

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

Diffstat

```
-rw-r--r-- drivers/iommu/iommu.c 3
```

1 files changed, 3 insertions, 0 deletions

diff --git a/drivers/iommu/iommu.c b/drivers/iommu/iommu.c

index 83c8e617a2c588..13c2a00f8aac1f 100644

--- a/drivers/iommu/iommu.c

+++ b/drivers/iommu/iommu.c

```
@@ -503,6 +503,9 @@ static void iommu_deinit_device(struct device *dev)
    dev->iommu_group = NULL;
    module_put(ops->owner);
    dev_iommu_free(dev);
+#ifdef CONFIG_IOMMU_DMA
+    dev->dma_iommu = false;
+#endif
}
```

```
DEFINE_MUTEX(iommu_probe_device_lock);
```