author	Ryo Takakura <ryotkkr98@gmail.com>	2025-02-18 09:08:30 +0100
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-04-25 10:45:19 +0200
commit	13e5148f70e81991acbe0bab5b1b50ba699116e7 (patch)	
tree	a9c83ed157de3161e75e09c0bb5eaee7721e466b	
parent	f8693e1bae9c08233a2f535c3f412e157df32b33 (diff)	
download	linux-13e5148f70e81991acbe0bab5b1b50ba699116e7.tar.gz	

diff options

context: 3

space: include

mode: unified

PCI: vmd: Make vmd_dev::cfg_lock a raw_spinlock_t type

```
[ Upstream commit 18056a48669a040bef491e63b25896561ee14d90 ]

The access to the PCI config space via pci_ops::read and pci_ops::write is
a low-level hardware access. The functions can be accessed with disabled
interrupts even on PREEMPT_RT. The pci_lock is a raw_spinlock_t for this
purpose.

A spinlock_t becomes a sleeping lock on PREEMPT_RT, so it cannot be
acquired with disabled interrupts. The vmd_dev::cfg_lock is accessed in
the same context as the pci_lock.

Make vmd_dev::cfg_lock a raw_spinlock_t type so it can be used with
interrupts disabled.

This was reported as:

  BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48
  Call Trace:
   rt_spin_lock+0x4e/0x130
   vmd_pci_read+0x8d/0x100 [vmd]
   pci_user_read_config_byte+0x6f/0xe0
   pci_read_config+0xfe/0x290
   sysfs_kf_bin_read+0x68/0x90

Signed-off-by: Ryo Takakura <ryotkkr98@gmail.com>
Tested-by: Luis Claudio R. Goncalves <lgoncalv@redhat.com>
Acked-by: Luis Claudio R. Goncalves <lgoncalv@redhat.com>
[bigeasy: reword commit message]
Signed-off-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
Tested-off-by: Luis Claudio R. Goncalves <lgoncalv@redhat.com>
Link: https://lore.kernel.org/r/20250218080830.ufw3IgyX@linutronix.de
[kwilczynski: commit log]
Signed-off-by: Krzysztof Wilczyński <kwilczynski@kernel.org>
[bhelgaas: add back report info from
https://lore.kernel.org/lkml/20241218115951.83062-1-ryotkkr98@gmail.com/]
Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- drivers/pci/controller/vmd.c 12

1 files changed, 6 insertions, 6 deletions

```
diff --git a/drivers/pci/controller/vmd.c b/drivers/pci/controller/vmd.c
index 5ff2066aa51643..dfa222e02c4da9 100644
--- a/drivers/pci/controller/vmd.c
+++ b/drivers/pci/controller/vmd.c
@@ -125,7 +125,7 @@ struct vmd_irq_list {
 struct vmd_dev {
        struct pci_dev          *dev;

-       spinlock_t              cfg_lock;
+       raw_spinlock_t          cfg_lock;
        void __iomem            *cfgbar;

        int msix_count;
@@ -402,7 +402,7 @@ static int vmd_pci_read(struct pci_bus *bus, unsigned int devfn, int reg,
        if (!addr)
                return -EFAULT;

-       spin_lock_irqsave(&vmd->cfg_lock, flags);
+       raw_spin_lock_irqsave(&vmd->cfg_lock, flags);
        switch (len) {
        case 1:
                *value = readb(addr);
@@ -417,7 +417,7 @@ static int vmd_pci_read(struct pci_bus *bus, unsigned int devfn, int reg,
                ret = -EINVAL;
                break;
        }
-       spin_unlock_irqrestore(&vmd->cfg_lock, flags);
+       raw_spin_unlock_irqrestore(&vmd->cfg_lock, flags);
        return ret;
 }

@@ -437,7 +437,7 @@ static int vmd_pci_write(struct pci_bus *bus, unsigned int devfn, int reg,
        if (!addr)
                return -EFAULT;

-       spin_lock_irqsave(&vmd->cfg_lock, flags);
+       raw_spin_lock_irqsave(&vmd->cfg_lock, flags);
        switch (len) {
        case 1:
                writeb(value, addr);
@@ -455,7 +455,7 @@ static int vmd_pci_write(struct pci_bus *bus, unsigned int devfn, int reg,
                ret = -EINVAL;
                break;
        }
-       spin_unlock_irqrestore(&vmd->cfg_lock, flags);
+       raw_spin_unlock_irqrestore(&vmd->cfg_lock, flags);
        return ret;
 }

@@ -1020,7 +1020,7 @@ static int vmd_probe(struct pci_dev *dev, const struct pci_device_id *id)
        if (features & VMD_FEAT_OFFSET_FIRST_VECTOR)
                vmd->first_vec = 1;

-       spin_lock_init(&vmd->cfg_lock);
+       raw_spin_lock_init(&vmd->cfg_lock);
        pci_set_drvdata(dev, vmd);
        err = vmd_enable_domain(vmd, features);
        if (err)
```