

## Re: 2.6.24-rt1 IRQ routing anomaly

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-02/msg10998.html>

---

- *From:* Mark Hounschell <[markh@xxxxxxxxxx](mailto:markh@xxxxxxxxxx)>
  - *Date:* Thu, 21 Feb 2008 08:30:49 -0500
- 

Steven Rostedt wrote:

[CC'd Thomas and Jon]

Thomas, Jon, looks like the someone has the funny interrupt controller.

On Thu, 21 Feb 2008, Mark Hounschell wrote:

According to /proc/interrupts, every interrupt received by eth1 is also being received by the sound card EMU10K1. The problem showed itself first with this. The sound system was quiet BTW.

It does not happen with 2.6.24 vanilla.

```
kernel: irq 19: nobody cared (try booting with the "irqpoll" option)
kernel: Pid: 1832, comm: IRQ-19 Not tainted 2.6.24.2-crt #2
kernel: [] __report_bad_irq+0x36/0x75
kernel: [] note_interrupt+0x1f7/0x227
kernel: [] thread_simple_irq+0x61/0x74
kernel: [] do_irqd+0x0/0x22f
kernel: [] do_irqd+0xb2/0x22f
kernel: [] do_irqd+0x0/0x22f
kernel: [] kthread+0x38/0x5d
kernel: [] kthread+0x0/0x5d
kernel: [] kernel_thread_helper+0x7/0x10
kernel: =====
kernel: -----
kernel: | preempt count: 00000001 |
kernel: | 1-level deep critical section nesting:
kernel: -----
kernel: .. [] .... __spin_lock_irq+0xe/0x1e
kernel: ....[<00000000>] .. ( <= _stext+0x3feff00/0x14)
kernel:
kernel: handlers:
kernel: [
```

Re: 2.6.24-rt1 IRQ routing anomaly

```
kernel: [<c013d6da>] __report_bad_irq+0x36/0x75
kernel: [<c013d910>] note_interrupt+0x1f7/0x227
kernel: [<c013ce85>] thread_simple_irq+0x61/0x74
kernel: [<c013d455>] do_irqd+0x0/0x22f
kernel: [<c013d507>] do_irqd+0xb2/0x22f
kernel: [<c013d455>] do_irqd+0x0/0x22f
kernel: [<c012b137>] kthread+0x38/0x5d
kernel: [<c012b0ff>] kthread+0x0/0x5d
kernel: [<c0104c13>] kernel_thread_helper+0x7/0x10
kernel: =====
kernel: -----
kernel: | preempt count: 00000001 |
kernel: | 1-level deep critical section nesting:
kernel: -----
kernel: .. [<c02b03b3>] .... __spin_lock_irq+0xe/0x1e
kernel: .. [<00000000>] .. ( <= _stext+0x3feff00/0x14)
kernel:
kernel: handlers:
kernel: [<f4d16544>] (snd_emu10k1_interrupt+0x0/0x42c [snd_emu10k1])
```

Looking at /proc/interrupts I could see the the EMU10K1 interrupt was going to town. I was busy busy on eth1 at the time.

So a simple external ping test with a quiet system at run level-3 revealed:

```
# lspci cat before.ping
CPU0 CPU1
0: 85 0 IO-APIC-edge timer
1: 396 420 IO-APIC-edge i8042
3: 4 2 IO-APIC-edge
4: 5 1 IO-APIC-edge
6: 1 4 IO-APIC-edge floppy
7: 0 0 IO-APIC-edge parport0
8: 2 0 IO-APIC-edge rtc
9: 0 1 IO-APIC-fasteoi acpi
12: 21 84 IO-APIC-edge i8042
14: 8457 8179 IO-APIC-edge libata
15: 1016 1519 IO-APIC-edge libata
16: 60 60 IO-APIC-fasteoi aic7xxx
17: 113 96 IO-APIC-fasteoi eth1
18: 44 47 IO-APIC-fasteoi
19: 99 114 IO-APIC-fasteoi EMU10K1
NMI: 0 0 Non-maskable interrupts
LOC: 93895 94157 Local timer interrupts
RES: 8831 8188 Rescheduling interrupts
CAL: 4176 5267 function call interrupts
TLB: 271 235 TLB shootdowns
TRM: 0 0 Thermal event interrupts
SPU: 0 0 Spurious interrupts
ERR: 0
MIS: 0
```

## Re: 2.6.24-rt1 IRQ routing anomaly

Then from an external machine: ping -c10 10.10.10.200

```
# cat after.ping
CPU0 CPU1
0: 85 0 IO-APIC-edge timer
1: 464 432 IO-APIC-edge i8042
3: 4 2 IO-APIC-edge
4: 5 1 IO-APIC-edge
6: 1 4 IO-APIC-edge floppy
7: 0 0 IO-APIC-edge parport0
8: 2 0 IO-APIC-edge rtc
9: 0 1 IO-APIC-fasteoi acpi
12: 21 84 IO-APIC-edge i8042
14: 8460 8198 IO-APIC-edge libata
15: 1360 1549 IO-APIC-edge libata
16: 60 60 IO-APIC-fasteoi aic7xxx
17: 129 102 IO-APIC-fasteoi eth1
18: 44 47 IO-APIC-fasteoi
19: 105 130 IO-APIC-fasteoi EMU10K1
NMI: 0 0 Non-maskable interrupts
LOC: 104387 104637 Local timer interrupts
RES: 8890 8214 Rescheduling interrupts
CAL: 4176 5267 function call interrupts
TLB: 271 236 TLB shootdowns
TRM: 0 0 Thermal event interrupts
SPU: 0 0 Spurious interrupts
ERR: 0
MIS: 0
```

44 interrupts added to both eth1 and EMU10K1

This is a known problem with this. Some interrupt controllers are funny and do funny things when an interrupt is masked, but interrupts enabled. They route the interrupt to the wrong interrupt line. The only reason that vanilla doesn't show it, is that vanilla does the interrupt handler when the interrupt is triggered, so it has no need to mask. RT on the other hand, runs interrupts in threaded context, which triggers this little quirk because we mask the interrupt. For some strange reason, the interrupt controller will trigger the interrupt for another interrupt, if that interrupt line is masked.

To prove this is the problem, boot with noapic in the kernel command line.

- 1) the problem should disappear.
- 2) (I'm betting) you see that the eth and EMU10K1 share the same interrupt line.

Re: 2.6.24-rt1 IRQ routing anomaly

Yep, you were right. They do share the same IRQ and the problem does go away. Unfortunately I can't run this machine with noapic. I need irq affinity.

I see from the back trace that this is i386. We have a workaround for this on x86\_64. Jon Masters has been working on better solutions too.

Yes this is i386. Is it just a certain interrupt controller that acts this way or are there more? I guess I have at least one of "those funny interrupt controllers" Hmm

```
#lspci
```

```
00:00.0 Host bridge: Advanced Micro Devices [AMD] AMD-760 MP
[IGD4-2P]
System Controller (rev 20)
00:01.0 PCI bridge: Advanced Micro Devices [AMD] AMD-760 MP
[IGD4-2P]
AGP Bridge
00:07.0 ISA bridge: Advanced Micro Devices [AMD] AMD-768 [Opus]
ISA (rev 05)
00:07.1 IDE interface: Advanced Micro Devices [AMD] AMD-768 [Opus]
IDE
(rev 04)
00:07.3 Bridge: Advanced Micro Devices [AMD] AMD-768 [Opus] ACPI
(rev 03)
00:08.0 Ethernet controller: 3Com Corporation 3c905C-TX/TX-M
[Tornado]
(rev 6c)
00:09.0 Class Class ff00: Compro Computer Services, Inc. Unknown device
4610 (rev 03)
00:10.0 PCI bridge: Advanced Micro Devices [AMD] AMD-768 [Opus] PCI
(rev 05)
01:05.0 VGA compatible controller: nVidia Corporation NV25 [GeForce4 Ti
4400] (rev a2)
02:04.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev
04)
02:04.1 Input device controller: Creative Labs SB Live! Game Port (rev 01)
02:05.0 Communication controller: National Instruments PCI-GPIB (rev 01)
02:06.0 SCSI storage controller: Adaptec AHA-2930CU (rev 03)
02:07.0 Communication controller: National Instruments PCI-GPIB (rev 01)
02:08.0 Ethernet controller: 3Com Corporation 3c905C-TX/TX-M
[Tornado]
(rev 78)
```

Again this does not happen with 2.6.24 vanilla. I'm not sure about

Re: 2.6.24-rt1 IRQ routing anomaly

earlier RT kernels.

Regards  
Mark

--

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>