

How do I debug PCI resource allocation problems

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-11/msg02648.html>

- *From:* Rainer Koenig <Rainer.Koenig@xxxxxx>
 - *Date:* Thu, 8 Nov 2007 20:13:42 +0100
-

This will get long, sorry. But I'm a bit desperate because I encounter strange problems on a new mainboard with Intel Q35 chipset and a shared memory graphics card. The logs and data I use here are from a SLED10 SP1 (x86_64) installation, but the problem occurs whatever distribution I try out.

Ok, short description of the problem:

I run 64-bit Linux using 2 GB of RAM, no problem at all. Then I turn off the machine, add 2 more GB so that now I have 4 GB of RAM. Turning it on I see the splashscreen of the boot loader, starting the kernel turns the screen black and that's it. The machine comes up, I can even ssh to it over the net. That is how I obtained the following data.

First a look at the boot.msg log. I will point out places of interest (at least the lines I think that are important for further analysis).

```
-----8<-boot.msg--start-----
Inspecting /boot/System.map-2.6.16.46-0.12-smp
Loaded 23173 symbols from /boot/System.map-2.6.16.46-0.12-smp.
Symbols match kernel version 2.6.16.
No module symbols loaded - kernel modules not enabled.

klogd 1.4.1, log source = ksyslog started.
<4>Bootdata ok (command line is
root=/dev/disk/by-id/scsi-SATA_ST3160815AS_9RX01AP0-part5 vga=0x31a
resume=/dev/sda1 splash=silent)
<5>Linux version 2.6.16.46-0.12-smp (geeko@buildhost) (gcc version 4.1.2
20070115 (prerelease) (SUSE Linux)) #1 SMP Thu May 17 14:00:09 UTC 2007
<6>BIOS-provided physical RAM map:
<4> BIOS-e820: 0000000000000000 - 000000000009d800 (usable)
<4> BIOS-e820: 000000000009d800 - 00000000000a0000 (reserved)
<4> BIOS-e820: 00000000000ce000 - 00000000000d0000 (reserved)
<4> BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
<4> BIOS-e820: 0000000000100000 - 00000000df5d0000 (usable)
<4> BIOS-e820: 00000000df5d0000 - 00000000df5dc000 (ACPI data)
<4> BIOS-e820: 00000000df5dc000 - 00000000df5df000 (ACPI NVS)
<4> BIOS-e820: 00000000df5df000 - 00000000df700000 (reserved)
<4> BIOS-e820: 00000000df800000 - 00000000e0100000 (reserved)
<4> BIOS-e820: 00000000f8000000 - 00000000fc000000 (reserved)
```

How do I debug PCI resource allocation problems

```
<4> BIOS-e820: 00000000fec00000 – 00000000fec10000 (reserved)
<4> BIOS-e820: 00000000fee00000 – 00000000fee01000 (reserved)
<4> BIOS-e820: 00000000ffb00000 – 0000000100000000 (reserved)
```

***** Comment: The following 2 lines are added when I upgrade from 2 GB to 4 GB.

```
<4> BIOS-e820: 0000000100000000 – 000000011a000000 (usable)
<4> BIOS-e820: 000000011a000000 – 000000011c000000 (reserved)
<6>DMI present.
<7>ACPI: RSDP (v000 PTLTD ) @
0x000000000000f7240
<7>ACPI: RSDT (v001 PTLTD RSDT 0x00060000 LTP 0x00000000) @
0x00000000df5d6d0b
<7>ACPI: FADT (v001 FSC 0x00060000 0x000f4240) @
0x00000000df5dba5f
<7>ACPI: TPCA (v001 Phoeni x 0x00060000 TL 0x00000000) @
0x00000000df5dbad3
<7>ACPI: _MAR (v001 Intel OEMDMAR 0x00060000 LOHR 0x00000001) @
0x00000000df5dbb05
<7>ACPI: SSDT (v001 FSC PST_CPU0 0x00060000 CSF 0x00000001) @
0x00000000df5dbb35
<7>ACPI: SSDT (v001 FSC PST_CPU1 0x00060000 CSF 0x00000001) @
0x00000000df5dbbeb
<7>ACPI: SSDT (v001 FSC PST_CPU2 0x00060000 CSF 0x00000001) @
0x00000000df5dbca1
<7>ACPI: SSDT (v001 FSC PST_CPU3 0x00060000 CSF 0x00000001) @
0x00000000df5dbd57
<7>ACPI: SPCR (v001 PTLTD $UCRTBLS$ 0x00060000 PTL 0x00000001) @
0x00000000df5dbe0d
<7>ACPI: MCFG (v001 PTLTD MCFG 0x00060000 LTP 0x00000000) @
0x00000000df5dbe5d
<7>ACPI: HPET (v001 PTLTD HPETTBL 0x00060000 LTP 0x00000001) @
0x00000000df5dbe99
<7>ACPI: MADT (v001 PTLTD APIC 0x00060000 LTP 0x00000000) @
0x00000000df5dbed1
<7>ACPI: BOOT (v001 PTLTD $SBFTBLS$ 0x00060000 LTP 0x00000001) @
0x00000000df5dbf55
<7>ACPI: ASF! (v016 CETP CETP 0x00060000 PTL 0x00000001) @
0x00000000df5dbf7d
<7>ACPI: DSDT (v001 FSC D2587/A1 0x00060000 MSFT 0x03000001) @
0x0000000000000000
<6>No NUMA configuration found
<6>Faking a node at 0000000000000000–000000011a000000
<6>Bootmem setup node 0 0000000000000000–000000011a000000
<7>On node 0 totalpages: 1004539
<7> DMA zone: 2979 pages, LIFO batch:0
<7> DMA32 zone: 896520 pages, LIFO batch:31
<7> Normal zone: 105040 pages, LIFO batch:31
<7> HighMem zone: 0 pages, LIFO batch:0
<6>ACPI: PM–Timer IO Port: 0x1008
```

How do I debug PCI resource allocation problems

```
<7>ACPI: Local APIC address 0xfe00000
<6>ACPI: LAPIC (acpi_id[0x00] lapic_id[0x00] enabled)
<6>Processor #0 6:15 APIC version 20
<6>ACPI: LAPIC (acpi_id[0x01] lapic_id[0x01] enabled)
<6>Processor #1 6:15 APIC version 20
<6>ACPI: LAPIC (acpi_id[0x02] lapic_id[0x02] enabled)
<6>Processor #2 6:15 APIC version 20
<6>ACPI: LAPIC (acpi_id[0x03] lapic_id[0x03] enabled)
<6>Processor #3 6:15 APIC version 20
<6>ACPI: LAPIC_NMI (acpi_id[0x00] high edge lint[0x1])
<6>ACPI: LAPIC_NMI (acpi_id[0x01] high edge lint[0x1])
<6>ACPI: LAPIC_NMI (acpi_id[0x02] high edge lint[0x1])
<6>ACPI: LAPIC_NMI (acpi_id[0x03] high edge lint[0x1])
<6>ACPI: IOAPIC (id[0x04] address[0xfec00000] gsi_base[0])
<6>IOAPIC[0]: apic_id 4, version 32, address 0xfec00000, GSI 0-23
<6>ACPI: INT_SRC_OVR (bus 0 bus_irq 0 global_irq 2 high edge)
<6>ACPI: INT_SRC_OVR (bus 0 bus_irq 9 global_irq 9 high level)
<7>ACPI: IRQ0 used by override.
<7>ACPI: IRQ2 used by override.
<7>ACPI: IRQ9 used by override.
<6>Setting APIC routing to physical flat
<6>ACPI: HPET id: 0xffffffff base: 0xfed00000
<6>Using ACPI (MADT) for SMP configuration information
<6>Allocating PCI resources starting at e2000000 (gap: e0100000:17f00000)
<6>SMP: Allowing 4 CPUs, 0 hotplug CPUs
<4>Built 1 zonelists
<5>Kernel command line:
root=/dev/disk/by-id/scsi-SATA_ST3160815AS_9RX01AP0-part5 vga=0x31a
resume=/dev/sda1 splash=silent
<6>boot splash: silent mode.
<4>Initializing CPU#0
<4>PID hash table entries: 4096 (order: 12, 131072 bytes)
<6>time.c: Using 14.318180 MHz WALL HPET GTOD HPET/TSC timer.
<6>time.c: Detected 2394.006 MHz processor.
<4>Console: colour dummy device 80x25
<4>Dentry cache hash table entries: 524288 (order: 10, 4194304 bytes)
<4>Inode-cache hash table entries: 262144 (order: 9, 2097152 bytes)
<4>Checking aperture...
<6>PCI-DMA: Using software bounce buffering for IO (SWIOTLB)
<6>Placing software IO TLB between 0x166f000 - 0x566f000
<4>Nosave address range: 000000000009d000 - 000000000009e000
<4>Nosave address range: 000000000009e000 - 00000000000a0000
<4>Nosave address range: 00000000000a0000 - 00000000000ce000
<4>Nosave address range: 00000000000ce000 - 00000000000d0000
<4>Nosave address range: 00000000000d0000 - 00000000000e0000
<4>Nosave address range: 00000000000e0000 - 0000000000100000
<4>Nosave address range: 00000000df5d0000 - 00000000df5dc000
<4>Nosave address range: 00000000df5dc000 - 00000000df5df000
<4>Nosave address range: 00000000df5df000 - 00000000df700000
<4>Nosave address range: 00000000df700000 - 00000000df800000
<4>Nosave address range: 00000000df800000 - 00000000e0100000
```

How do I debug PCI resource allocation problems

```
<4>Nosave address range: 00000000e0100000 – 00000000f8000000
<4>Nosave address range: 00000000f8000000 – 00000000fc000000
<4>Nosave address range: 00000000fc000000 – 00000000fec00000
<4>Nosave address range: 00000000fec00000 – 00000000fec10000
<4>Nosave address range: 00000000fec10000 – 00000000fee00000
<4>Nosave address range: 00000000fee00000 – 00000000fee01000
<4>Nosave address range: 00000000fee01000 – 00000000ffb00000
<4>Nosave address range: 00000000ffb00000 – 0000000100000000
<4>Memory: 3942960k/4620288k available (1918k kernel code, 142212k reserved,
886k data, 196k init)
<4>Calibrating delay using timer specific routine.. 4792.10 BogoMIPS
(lpj=9584204)
<6>Security Framework v1.0.0 initialized
<4>Mount-cache hash table entries: 256
<6>CPU: L1 I cache: 32K, L1 D cache: 32K
<6>CPU: L2 cache: 4096K
<4>using mwait in idle threads.
<6>CPU: Physical Processor ID: 0
<6>CPU: Processor Core ID: 0
<6>CPU0: Thermal monitoring enabled (TM2)
<6>checking if image is initramfs... it is
<4>Freeing initrd memory: 2638k freed
<4> not found!
<6>activating NMI Watchdog ... done.
<6>Using local APIC timer interrupts.
<4>result 16625020
<6>Detected 16.625 MHz APIC timer.
<6>Booting processor 1/4 APIC 0x1
<4>Initializing CPU#1
<4>Calibrating delay using timer specific routine.. 4788.04 BogoMIPS
(lpj=9576089)
<6>CPU: L1 I cache: 32K, L1 D cache: 32K
<6>CPU: L2 cache: 4096K
<6>CPU: Physical Processor ID: 0
<6>CPU: Processor Core ID: 1
<6>CPU1: Thermal monitoring enabled (TM2)
<4>Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz stepping 0b
<6>Booting processor 2/4 APIC 0x2
<4>Initializing CPU#2
<4>Calibrating delay using timer specific routine.. 4788.10 BogoMIPS
(lpj=9576217)
<6>CPU: L1 I cache: 32K, L1 D cache: 32K
<6>CPU: L2 cache: 4096K
<6>CPU: Physical Processor ID: 0
<6>CPU: Processor Core ID: 2
<6>CPU2: Thermal monitoring enabled (TM2)
<4>Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz stepping 0b
<6>Booting processor 3/4 APIC 0x3
<4>Initializing CPU#3
<4>Calibrating delay using timer specific routine.. 4788.10 BogoMIPS
(lpj=9576210)
```

How do I debug PCI resource allocation problems

```
<6>CPU: L1 I cache: 32K, L1 D cache: 32K
<6>CPU: L2 cache: 4096K
<6>CPU: Physical Processor ID: 0
<6>CPU: Processor Core ID: 3
<6>CPU3: Thermal monitoring enabled (TM2)
<4>Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz stepping 0b
<6>Brought up 4 CPUs
<6>testing NMI watchdog ... OK.
<4>migration_cost=10,3344
<6>NET: Registered protocol family 16
<6>ACPI: bus type pci registered
<6>PCI: Using configuration type 1
<3>PCI: BIOS Bug: MCFG area is not E820-reserved
<3>PCI: Not using MMCONFIG.
<6>ACPI: Subsystem revision 20060127
<6>ACPI: Interpreter enabled
<6>ACPI: Using IOAPIC for interrupt routing
<6>ACPI: PCI Root Bridge [PCI0] (0000:00)
<7>PCI: Probing PCI hardware (bus 00)
<7>Boot video device is 0000:00:02.0
<6>PCI: Ignoring BAR0-3 of IDE controller 0000:00:1f.2
<6>PCI: Transparent bridge - 0000:00:1e.0
<7>ACPI: PCI Interrupt Routing Table [_SB_.PCI0._PRT]
<7>ACPI: PCI Interrupt Routing Table [_SB_.PCI0.PEXA._PRT]
<7>ACPI: PCI Interrupt Routing Table [_SB_.PCI0.PEXB._PRT]
<7>ACPI: PCI Interrupt Routing Table [_SB_.PCI0.PEXC._PRT]
<7>ACPI: PCI Interrupt Routing Table [_SB_.PCI0.PCIH._PRT]
<4>ACPI: PCI Interrupt Link [LNKA] (IRQs 3 4 7 9 10 *11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKB] (IRQs 3 4 7 9 10 11 12 14 15) *5
<4>ACPI: PCI Interrupt Link [LNKC] (IRQs 3 4 7 9 *10 11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKD] (IRQs 3 4 7 9 10 *11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKE] (IRQs 3 4 7 *9 10 11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKF] (IRQs 3 4 7 9 10 *11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKG] (IRQs 3 4 7 9 10 *11 12 14 15)
<4>ACPI: PCI Interrupt Link [LNKH] (IRQs 3 4 7 9 10 *11 12 14 15)
<4>ACPI: Device [ECP] status [00000008]: functional but not present; setting
present
<4>ACPI: Device [COM2] status [00000008]: functional but not present; setting
present
<6>PCI: Using ACPI for IRQ routing
<6>PCI: If a device doesn't work, try "pci=routeirq". If it helps, post a
report
<3>PCI: Cannot allocate resource region 7 of bridge 0000:00:01.0
<3>PCI: Cannot allocate resource region 7 of bridge 0000:00:1c.0
<3>PCI: Cannot allocate resource region 7 of bridge 0000:00:1c.4
<3>PCI: Cannot allocate resource region 2 of device 0000:00:02.0
```

***** Question: How do I get more info on the messages above? How can I debug that and find a root cause for the messages? 0000:00:02.0 is the VGA device that is giving me those headaches BTW.

How do I debug PCI resource allocation problems

```
<6>hpet0: at MMIO 0xfed00000 (virtual 0xffffffff5fe000), IRQs 2, 8, 0, 0
<6>hpet0: 4 64-bit timers, 14318180 Hz
<4>PCI: Ignore bogus resource 6 [0:0] of 0000:00:02.0
<6>PCI: Bridge: 0000:00:01.0
<6> IO window: disabled.
<6> MEM window: disabled.
<6> PREFETCH window: disabled.
<6>PCI: Bridge: 0000:00:1c.0
<6> IO window: disabled.
<6> MEM window: disabled.
<6> PREFETCH window: disabled.
<6>PCI: Bridge: 0000:00:1c.4
<6> IO window: disabled.
<6> MEM window: disabled.
<6> PREFETCH window: disabled.
<6>PCI: Bridge: 0000:00:1e.0
<6> IO window: disabled.
<6> MEM window: disabled.
<6> PREFETCH window: disabled.
<6>GSI 16 sharing vector 0xA9 and IRQ 16
<6>ACPI: PCI Interrupt 0000:00:01.0[A] -> GSI 16 (level, low) -> IRQ 169
<7>PCI: Setting latency timer of device 0000:00:01.0 to 64
<6>GSI 17 sharing vector 0xB1 and IRQ 17
<6>ACPI: PCI Interrupt 0000:00:1c.0[A] -> GSI 18 (level, low) -> IRQ 177
<7>PCI: Setting latency timer of device 0000:00:1c.0 to 64
<6>GSI 18 sharing vector 0xB9 and IRQ 18
<6>ACPI: PCI Interrupt 0000:00:1c.4[B] -> GSI 22 (level, low) -> IRQ 185
<7>PCI: Setting latency timer of device 0000:00:1c.4 to 64
<7>PCI: Setting latency timer of device 0000:00:1e.0 to 64
<6>Simple Boot Flag at 0x43 set to 0x1
<4>IA32 emulation $Id: sys_ia32.c,v 1.32 2002/03/24 13:02:28 ak Exp $
<6>audit: initializing netlink socket (disabled)
<5>audit(1194361547.452:1): initialized
<4>Total HugeTLB memory allocated, 0
<5>VFS: Disk quotas dquot_6.5.1
<4>Dquot-cache hash table entries: 512 (order 0, 4096 bytes)
<6>Initializing Cryptographic API
<6>io scheduler noop registered
<6>io scheduler anticipatory registered
<6>io scheduler deadline registered
<6>io scheduler cfq registered (default)
<6>ACPI: PCI Interrupt 0000:00:01.0[A] -> GSI 16 (level, low) -> IRQ 169
<7>PCI: Setting latency timer of device 0000:00:01.0 to 64
<4>assign_interrupt_mode Found MSI capability
<7>Allocate Port Service[0000:00:01.0:pcie00]
<6>ACPI: PCI Interrupt 0000:00:1c.0[A] -> GSI 18 (level, low) -> IRQ 177
<7>PCI: Setting latency timer of device 0000:00:1c.0 to 64
<4>assign_interrupt_mode Found MSI capability
<7>Allocate Port Service[0000:00:1c.0:pcie00]
<7>Allocate Port Service[0000:00:1c.0:pcie02]
<6>ACPI: PCI Interrupt 0000:00:1c.4[B] -> GSI 22 (level, low) -> IRQ 185
```

How do I debug PCI resource allocation problems

```
<7>PCI: Setting latency timer of device 0000:00:1c.4 to 64
<4>assign_interrupt_mode Found MSI capability
<7>Allocate Port Service[0000:00:1c.4:pcie00]
<7>Allocate Port Service[0000:00:1c.4:pcie02]
<4>vesafb: cannot reserve video memory at 0xe0000000
```

***** Comment: I guess this is why I get a black screen. But why can't that memory be reserved?

```
<6>vesafb: framebuffer at 0xe0000000, mapped to 0xffffc20000080000, using
8128k, total 8128k
```

***** Comment: The framebuffer address with 0xfff... looks pretty strange, doesn't it?

```
<6>vesafb: mode is 1280x1024x16, linelength=2560, pages=2
<6>vesafb: scrolling: redraw
<6>vesafb: Truecolor: size=0:5:6:5, shift=0:11:5:0
<6>bootsplash 3.1.6-2004/03/31: looking for picture...<6> silentjpeg size
65383 bytes,<6>...found (1280x1024, 55554 bytes, v3).
<4>Console: switching to colour frame buffer device 155x60
<6>fb0: VESA VGA frame buffer device
<6>Real Time Clock Driver v1.12ac
<7>hpet_resources: 0xfed00000 is busy
<6>Non-volatile memory driver v1.2
<6>Linux agpgart interface v0.101 (c) Dave Jones
<6>serio: i8042 AUX port at 0x60,0x64 irq 12
<6>serio: i8042 KBD port at 0x60,0x64 irq 1
<6>Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
<6>serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
<6>GSI 19 sharing vector 0xE1 and IRQ 19
<6>ACPI: PCI Interrupt 0000:00:03.3[B] -> GSI 17 (level, low) -> IRQ 225
<6>0000:00:03.3: ttyS1 at I/O 0x1c90 (irq = 225) is a 16550A
<4>RAMDISK driver initialized: 16 RAM disks of 128000K size 1024 blocksize
<6>mice: PS/2 mouse device common for all mice
<6>input: AT Translated Set 2 keyboard as /class/input/input0
<6>input: PC Speaker as /class/input/input1
<6>md: md driver 0.90.3 MAX_MD_DEVS=256, MD_SB_DISKS=27
<6>md: bitmap version 4.39
<6>NET: Registered protocol family 2
<4>IP route cache hash table entries: 131072 (order: 8, 1048576 bytes)
<4>TCP established hash table entries: 262144 (order: 10, 4194304 bytes)
<4>TCP bind hash table entries: 65536 (order: 8, 1048576 bytes)
<6>TCP: Hash tables configured (established 262144 bind 65536)
<6>TCP reno registered
<6>NET: Registered protocol family 1
<4>ACPI wakeup devices:
<4>PEXA PEXX PEXB PEXC PEXD USB1 USB2 USB3 USB4 USB5 USB6 USB7 USB8 USB9 LAN
PCIH KEYB PS2M COM1 COM2
<6>ACPI: (supports S0 S1 S3 S4 S5)
<4>Freeing unused kernel memory: 196k freed
```

How do I debug PCI resource allocation problems

```
<6>Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
<6>ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
<5>SCSI subsystem initialized
<7>libata version 2.00 loaded.
<7>ata_piix 0000:00:1f.2: version 2.00ac7
<6>ata_piix 0000:00:1f.2: MAP [ P0 P2 P1 P3 ]
<6>GSI 20 sharing vector 0xE9 and IRQ 20
-----8<-boot.msg--end-----
```

Ok, so far so good. Looks like my problems come from the PCI resources of the VGA card. So I did a lspci on that device and got another surprise:

```
00:02.0 VGA compatible controller: Intel Corporation Integrated Graphics
Controller (rev 02) (prog-if 00 [VGA])
Subsystem: Fujitsu Siemens Computer GmbH Unknown device 10fc
Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping-
SERR- FastB2B-
Status: Cap+ 66MHz- UDF- FastB2B+ ParErr- DEVSEL=fast >TAbort- <TAbort-
<MAbort- >SERR- <PERR-
Latency: 0
Interrupt: pin A routed to IRQ 11
Region 0: Memory at f0100000 (32-bit, non-prefetchable) [size=512K]
Region 1: I/O ports at 1c70 [size=8]
Region 2: Memory at 120000000 (32-bit, prefetchable) [size=256M]
Region 3: Memory at f0000000 (32-bit, non-prefetchable) [size=1M]
Capabilities: [90] Message Signalled Interrupts: Mask- 64bit- Queue=0/0
Enable-
Address: 00000000 Data: 0000
Capabilities: [d0] Power Management version 2
Flags: PMEClk- DSI+ D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)
Status: D0 PME-Enable- DSel=0 DScale=0 PME-
00: 86 80 b2 29 07 00 90 00 02 00 00 03 00 00 00 00
10: 00 00 10 f0 71 1c 00 00 08 00 00 20 00 00 00 f0
20: 00 00 00 00 00 00 00 00 00 00 00 00 34 17 fc 10
30: 00 00 00 00 90 00 00 00 00 00 00 00 0b 01 00 00
40: 09 00 0b b1 62 00 a0 c8 46 04 16 00 00 00 00 00
50: 00 00 30 01 4b 03 00 00 00 00 00 00 00 00 80 df
60: 00 00 02 02 00 00 00 00 00 00 00 00 00 00 00 00
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
90: 05 d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0: 11 11 00 00 00 00 06 03 00 00 00 00 00 00 00 00
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
d0: 01 00 22 00 00 00 00 00 00 00 00 00 00 01 02 00
e0: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00
f0: 12 00 03 00 00 00 00 00 90 0f 03 00 d1 cd 5d df
```

Question: Memory region 2 at 120000000? That is beyond the 4GB boundary and the BIOS guys I know told me that every PCI IOMEM region should reside within the first 4 GBs! When running the machine with 2 GB only lspci output looks

How do I debug PCI resource allocation problems

like this for the VGA device:

```
00:02.0 Class 0300: 8086:29b2 (rev 02) (prog-if 00 [VGA])
Subsystem: 1734:10fc
Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping-
SERR- FastB2B-
Status: Cap+ 66MHz- UDF- FastB2B+ ParErr- DEVSEL=fast >TAbort- <TAbort-
<MAbort- >SERR- <PERR-
Latency: 0
Interrupt: pin A routed to IRQ 11
Region 0: Memory at f0100000 (32-bit, non-prefetchable) [size=512K]
Region 1: I/O ports at 1c70 [size=8]
Region 2: Memory at e0000000 (32-bit, prefetchable) [size=256M]
Region 3: Memory at f0000000 (32-bit, non-prefetchable) [size=1M]
Capabilities: [90] Message Signalled Interrupts: Mask- 64bit- Queue=0/0
Enable-
Address: 00000000 Data: 0000
Capabilities: [d0] Power Management version 2
Flags: PMEClk- DSI+ D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)
Status: D0 PME-Enable- DSel=0 DScale=0 PME-
```

That means now we get the region 2 at e0000000 and everything is fine.

I tried to have a look at the PCI config space with a DOS tool from the year 1999, the hexdump looked pretty much similar, but byte 1A changed from "20" to "e0". And the region was declared as e0000008, which also looked strange to me.

This is where I am now. I also with an Intel reference mainboard (same chipset, different BIOS) and this one didn't show the problem. That makes me think that the problem is somewhere in the BIOS, but where? I have access to the BIOS developers, but they don't know much about Linux and since the other operating systems from Redmond are running without problems they are hard to convince that they made a mistake. :-)

Another very bad side effect of the problem is that when the machine runs on 32-bit Linux then the graphic card seems to work, but people report corrupted file systems after a while. I guess that is related to my problem on 64 bit, only that in the 32-bit case then filesystem buffers got overwritten by the video RAM and when they are written back to disk... ouch!

I also tried to track the problem down with the CD from LinuxFirmwareKit.org, but the resource allocation errors are the same and unfortunately the lack of verbosity as well.

Ok, that's it. Any help is much appreciated.

Thanks

Rainer

--

Rainer Koenig, Diplom-Informatiker (FH), Augsburg, Germany

-

How do I debug PCI resource allocation problems

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>