

## Fixes for nforce2 hard lockup, apic, io-apic, udma133 covered

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-12/1528.html>

---

*From:* Ross Dickson ([ross\\_at\\_datcreative.com.au](mailto:ross_at_datcreative.com.au))

*Date:* 12/07/03

To: [linux-kernel@vger.kernel.org](mailto:linux-kernel@vger.kernel.org)  
Date: Sun, 7 Dec 2003 23:12:00 +1000

Greetings,

I am not subscribed so please cc responses.

I have monitored list and know my nforce2 experiences have been common.

Attached patches are in a single bzip tar ball.

I have Albatron KM18G Pro & Epox 8RGA+ MOBOs both using nforce2 chipsets.

I made up a kernel as follows.

```
Get std 2.4.22 src
apply patch-2.4.23
apply 2.4.22-low-latency.patch
apply preempt-kernel-rml-2.4.23-pre5-1.patch
apply vhz-j64-2.4.22.patch
```

One patch fails on inode.c, dispose\_list() so I placed conditional\_schedule() as follows

```
=static void dispose_list(struct list_head *head)
={
= int nr_disposed = 0;
=
= while (!list_empty(head)) {
= struct inode *inode;
= conditional_schedule();
```

Config for athlon with 1000hz tics, preempt & low-lat on.

Compiled and installed nvnet & nvidia video driver.

Disclaimer: The following information and code patches are not fully tested and may be dangerous, also these are the first patches I have made for public consumption so I hope that their format works.

Note also that the patches are against 2.4.22 even though they were developed against the heavily patched 2.4.23 mentioned above. The patch code is the same for both kernels but at different line numbers.

When I enabled either apic or io-apic in kern config, lockups came hard and fast. Particularly bad under hard disk load. Heaps of lost ints on irq7 in apic and ioapic mode.

## Linux-Kernel: Fixes for nforce2 hard lockup, apic, io-apic, udma133 covered

Lockups disappeared when I lowered the ide hda udma speed to mode 3 with hdparm so I went looking for answers which now follow.

There are three parts to this email.

- a) apic mods.
- b) io-apic mods
- c) ide driver mods

a) Lockups are due to too fast an apic acknowledge of apic timer int.  
Apic hard locked up the system – no nmi debug available.  
Fixed it by introducing a delay of at least 500ns into smp\_apic\_timer\_interrupt() just prior to ack\_APIC\_irq().  
See attached diff file "nforce2-apic.c-2.4.22.patch" for details.  
I have guessed at a suitable cpu speed dependent delay.  
Perhaps someone with AMD cpu docs (apic timing specs) & analyser tools could refine it.

Maybe nforce2 chipset really is very quick accessing ram in dual dimm mode?  
Or AMD 2200XP has a really slow APIC?

```
--- linux-2.4.22/arch/i386/kernel/apic.c 2003-06-14 00:51:29.000000000 +1000
+++ linux-2.4.22-rd/arch/i386/kernel/apic.c 2003-12-07 18:27:32.000000000 +1000
@@ -1078,6 +1078,15 @@
     */
     apic_timer_irqs[cpu]++;

+#ifdef CONFIG_MK7 && CONFIG_BLK_DEV_AMD74XX
+ /*
+ * on 2200XP & nforce2 chipset we need at least 500ns delay here
+ * to stop lockups with udma100 drive. try to scale delay time
+ * with cpu speed. Ross Dickson.
+ */
+ ndelay((cpu_khz >> 12)+200); /* don't ack too soon or hard lockup */
+#endif
+
+ /*
+ * NOTE! We'd better ACK the irq immediately,
+ * because timer handling can be slow.
```

b) I was also disappointed to see I could not have irq0 timer IO-APIC-edge.  
So I have fixed it too (tested on both my epox and albatron MOBOS).  
Firstly I found 8254 connected directly to pin 0 not pin 2 of io-apic.  
I have modified check\_timer() in io\_apic.c to trial connect pin and test for it after the existing test for connection to io-apic.  
See attached diff file nforce2-io-apic.c-2.4.22 for details.

```
--- linux-2.4.22/arch/i386/kernel/io_apic.c 2003-08-25 21:44:39.000000000 +1000
+++ linux-2.4.22-rd/arch/i386/kernel/io_apic.c 2003-12-07 18:40:40.000000000 +1000
@@ -1614,9 +1614,44 @@
     return;
 }
 clear_IO_APIC_pin(0, pin1);
```

## Linux–Kernel: Fixes for nforce2 hard lockup, apic, io–apic, udma133 covered

```
– printk(KERN_ERR "..MP–BIOS bug: 8254 timer not connected to IO–APIC\n");
+ printk(KERN_ERR "..MP–BIOS bug: 8254 timer not connected to IO–APIC pin%d\n",pin1);
}
```

```
+#ifdef CONFIG_ACPI_BOOT && CONFIG_X86_UP_IOAPIC
+ /* for nforce2 try vector 0 on pin0
+ * Note the io_apic_set_pci_routing call disables the 8259 irq 0
+ * so we must be connected directly to the 8254 timer if this works
+ * Note2: this violates the above comment re Subtle but works!
+ */
+ printk(KERN_INFO "..TIMER: Is timer irq0 connected to IOAPIC Pin0? ...\n");
+ if ( pin1 != -1 && nr_ioapics ) {
+ int saved_timer_ack = timer_ack;
+ /* next call also disables 8259 irq0 */
+ int result = io_apic_set_pci_routing ( 0, 0, 0, 0, 0);
+ /*
+ * Ok, does IRQ0 through the IOAPIC work?
+ */
+ unmask_IO_APIC_irq(0);
+ timer_ack = 0 ;
+ if (timer_irq_works()) {
+ if (nmi_watchdog == NMI_IO_APIC) {
+ disable_8259A_irq(0);
+ setup_nmi();
+ enable_8259A_irq(0);
+ check_nmi_watchdog();
+ }
+ printk(KERN_INFO "..TIMER: works OK on apic pin0 irq0\n" );
+ return;
+ }
+ /* failed */
+ timer_ack = saved_timer_ack;
+ clear_IO_APIC_pin(0, 0);
+ result = io_apic_set_pci_routing ( 0, pin1, 0, 0, 0);
+ printk(KERN_ERR "..MP–BIOS bug: 8254 timer not connected to IO–APIC Pin 0\n");
+ }
+#endif
+ /* end new stuff for nforce2 */
+
+     printk(KERN_INFO "...trying to set up timer (IRQ0) through the 8259A ... ");
+     if (pin2 != -1) {
+         printk("\n..... (found pin %d) ...", pin2);
```

c) Finally during my fault finding I merged A.Martins patches for the nforce2 IDE driver.

I note that the nforce2 address setup timing bits are different to the AMD ones.

I have assumed the nforce2 address timings apply to nforce and nforce3 chipsets.

I could be wrong so if someone with the nvidia docs could check it please.

I have also not tested it with anything but a WDC ata100 hard drive.

For info see attached patch files (I think pci ids are already in 2.4.23)

nforce2–amd74xx.c–2.4.22.patch, nforce2–amd74xx.h–2.4.22.patch, nforce2–pci\_ids.h–2.4.22.patch

Linux-Kernel: Fixes for nforce2 hard lockup, apic, io-apic, udma133 covered

Thanks  
Ross Dickson

—

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

---

- application/x-tbz attachment: [ross-diffs.tar.bz2](#)