Re: [Bug #11382] e1000e: 2.6.27−rc1 corrupts EEPROM/NVM

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2008−09/msg08890.html

- *From*: Jesse Barnes <jbarnes@xxxxxxxxxxxxxxx>
- *Date*: Thu, 25 Sep 2008 11:56:08 −0700

On Thursday, September 25, 2008 10:24 am Jiri Kosina wrote:

> On Thu, 25 Sep 2008, Frans Pop wrote:
>
> > Extra datapoint. As far as I've seen this problem has not yet been reported by any people running Debian. This could point to X.Org as Debian currently has 7.3 while I think the reports so far have been with 7.4.
>
> Yes, I think that xorg/xorg i915 driver/libdrm/GEM/whatever are the biggest suspect currently, according to the data that has been gathered so far.

We have confirmation that this isn't GEM related; according to the Novell bug at https://bugzilla.novell.com/show_bug.cgi?id=425480 people have hit the problem with kernels w/o GEM.

That doesn't rule out i915 (though I don't think any changes have gone in since 2.6.26 that would have caused this) or xf86−video−intel. It's possible that X is getting confused about BAR mappings somehow, resulting in a clobbered e1000e NVRAM, but why would the kernel version matter in that case? The only thing that comes to mind would be PAT...

Recent versions of the X drivers (using recent libpciaccess code) will try to map the resourceN_wc file in sysfs. It's possible that the map size we end up using is wrong, leading to the situation Dave described earlier where we map too much MMIO space.

> Still, what confuses me a little bit −− the EEPROM of the card is set to all 0xff, once the corruption happens. Isn't that a quite a coincidence, that bytes representing "nothing" in this context are used?

Presumably one has to write all ones to the EEPROM BAR of the e1000 device to see that pattern? Or is there some way of configuring the EEPROM such that

it'll fail to respond to read cycles resulting in all ones for every read
back (i.e. target abort)?

Jesse
−−