

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

Re: + edac-new-opteron-athlon64-memory-controller-driver added to -mm tree

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-07/msg01619.html>

- *From:* Andi Kleen <ak@xxxxxx>
 - *Date:* 6 Jul 2006 15:01:53 +0200
 - *Date:* Thu, 6 Jul 2006 15:01:53 +0200
-

On Thu, Jul 06, 2006 at 12:12:14AM -0600, Eric W. Biederman wrote:

I think if this conversation is going to make headway we need to step back a minute, and ask what makes sense to do an where and not get caught up the details of an implementation.

It's rather from my POV -

we already got implementations (most "big" architectures have already own advanced hardware error reporting systems)

The EDAC folks want to add another.

It's not clear what advantages it gives.

As far as I can see it is in many ways a step back to at least compared to the x86-64 code:

uses printk, adds tons of code in kernel that is better in user space, doesn't use SMBIOS, requires more CPU specific code instead of using portable MCE interfaces, is very complicated, ...

Obviously I'm biased on this, but I went through many of these mistakes already myself when going from the 2.4 to 2.6 MCE handlers.

I can see it still being used for old chipsets or chipsets that don't support machine checks for memory errors, but these should be mostly legacy.

- Which cpu address did the error happen at.
So we can kill the processes using that memory.
Although simply killing the entire machine appears acceptable.

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

- What is the chipsets idea of which DIMM the memory error occurred on. For bus based memory architectures like the opteron this is a chip select of the DIMM rank. For serial memory architectures this is some kind of bus address, but still useful for describing individual chips.
- What is the silk screen label on the motherboard that corresponds to the chip selects with problems.

If you look at the memory controller, and the associated error reporting registers (which are sometimes available in the machine check). There has always been enough information to determine the hardware address the memory controller knows the DIMM by.

Getting the address of the error is usually possible but not always and not always very reliably.

Mapping between the hardware address that the memory controller knows DIMMS by and the actual DIMMS themselves is actually pretty easy even if you don't have any motherboard information.

That's all supposed to be done by the standard machine check handlers.

I think EDAC just started because some older chipsets don't integrate error reporting into the standard machine checks.

But in newer systems which are the way forward you get it from standard MCEs, no need for special drivers anymore.

It is just a matter of plugging in DIMMS in different positions and seeing which DIMMS that the hardware currently sees. It's maybe half a days work on an unknown motherboard.

Sorry that's totally unrealistic for anybody outside a hardware vendor or perhaps a big supercomputing lab. Normal users don't want to sit down "half a day" with their new systems to figure out to what the DIMMs map.

It either has to "just work" or they won't be able to use it.

We need to figure out some way to do it automatically. While SMBIOS is not perfect, it is far better than any manual proposals

That said I know SMBIOS can be wrong, so allowing to overwrite

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

it makes sense. But requiring the users to do this by default is a complete non starter IMHO.

knows the DIMM by requires the reading of hardware registers, some that are not easily accessible to user space so a kernel driver tends to make sense, just to get the information.

Possibly we could just export that information and let the user space figure it out from there. But memory is a key system

You can do it completely in user space. See mcelog as proof.

And figuring out the channel in a lot of code etc. seems overkill to me – or at least i haven't gotten an explanation why it's better than just using the reported address.

component and hardware designers are very creative so coming up with a consistent model would be very hard. So far we

Yes the error reporting is still machine specific so far. Doing it generically would be good.

have had to improve our helper functions every couple of chipsets The other pieces to me seem much more fluid. Especially since EDAC does not yet export much if anything to user space except through printk's in any stable kernel.

Yes that's another issue. printks are not very good for this. That is why I went over to a specialized logging device.

As for the suggestion of using DMI as best as I can determine it suffers rather badly from the never ending creativity of the chipset developers and does not have a model that can describe what needs to happen for the current generation of chipset much less the bleeding edge ones. Which is besides the fact that the only thing that you can usually trust in DMI tables is the motherboard manufacturer.

I think you paint it worse than it is. Also there are no realistic alternatives that I can see. Requiring all users to do it by hand is it certainly not.

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree

I do think getting the motherboard id out of DMI provides a great key to build a memory controller hardware address to DIMM label lookup table. With EDAC we have been computing that information in user space and caching it kernel side so we could generate immediately useful print statements. Which is handy but probably not necessary.

Ok that is a proposal, but still won't cover most motherboards that are out there.

Having an override table for motherboards where DMI is known to be wrong certainly makes sense to me. But the default has to be DMI I think.

If there was such a table somewhere I would be happy to support it with mcelog.

But who would be willing to maintain such a table? It would be a lot of work.

I'm still optimistic though – if Linux starts to use this information more aggressively then there will be much pressure from customers at least on server level kit vendors who still get this wrong.

This won't help all the cheap desktop/laptop boards , but these tend to usually not have more than two DIMMs, so it's not that big an issue.

–Andi

–

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/>

Re: + edac-new-opteron-athlon64-memory-controller-driver.patch added to -mm tree