

Re: [PATCH] Remove process freezer from suspend to RAM pathway

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-07/msg01707.html>

- *From:* "Rafael J. Wysocki" <rjw@xxxxxxx>
 - *Date:* Fri, 6 Jul 2007 09:35:10 +0200
-

On Friday, 6 July 2007 05:59, Benjamin Herrenschmidt wrote:

How about a freezer whose job it is to "wait for pending hard interrupts to complete when we have already guaranteed that we won't get any more"? That part should be really **REALLY** easy. You don't need to care about either userspace processes or kernel threads at all. Specifically, Step 1 consists of:

Well, waiting for pending DMA and making sure to not trigger more activity is what driver suspend() is supposed to do. With the ability for simple drivers that can cope with it to just basically use a late_suspend(), called after IRQs are off, that basically does what you describe: wait for pending HW tasks to complete (polling) and turn the damn thing off.

Note that the later is really a shortcut for somewhat dump and directly accessible devices (PCI comes to mind). Things like USB has to use the "normal" mechanism of blocking IOs etc... at suspend(), at least, USB devices have to since the USB HC will not issue any new URBs. (And will return them with a nice error code which is a perfect way to deal with it in driver, been there, it works fine, once we fixed the races in the USB host code itself, which I think we pretty much did by now).

Scheduling and userspace are all still fully enabled in this scenario. Once all your devices are turned off, the only remaining running threads will be those which haven't done IO since the beginning of the suspend. We can then disable preemption, turn off the timer interrupts, and tell the other CPUs to park all their remaining threads in schedule() and sleep. Then we put the IRQ controller to sleep and go to sleep ourselves. If our driver model locking is sufficient to handle putting a parent device to sleep while threads are sleeping on a child device then there are exactly 0 problems.

Re: [PATCH] Remove process freezer from suspend to RAM pathway

What you propose is basically a slightly over-simplistic version of what I think (and Paulus think) should be done. We do need to do it via driver callbacks down the tree since only drivers can know how to deal with their DMA etc... and ordering need to be respected, but that's basically it.

And guess what ? It's what we do on powerbooks, and it works fine, without a freezer :-)

On powerbooks you disable the nonboot CPUs before suspending devices, which simplifies things `_a_ _lot_` in comparison with the general case.

If you additionally disable kernel preemption, then you don't need anything like the freezer anyway in that case (except for detecting situations in which the suspend process can deadlock with a task stuck in the D state holding a lock, but that's overkill).

Greetings,
Rafael

--

"Premature optimization is the root of all evil." – Donald Knuth

-

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/>