Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Wed, 16 Aug 2006 06:37:14 GMT
View Forum Message <> Reply to Message

Salyzyn, Mark wrote:
> Others calls in the driver to shost_for_each_device unlock the host_lock
> while in the loop, makes sense to do the same in that loop as well given
> that these actions are taken when the adapter is quiesced. I worry,
> though, completion of the commands with QUEUE_FULL may result in them
> being turned around immediately which could clutter up the list. Could
> you experiment with this change:
>
> static void adpt_fail_posted_scbs(adpt_hba* pHba)
> {
> struct scsi_cmnd* cmd = NULL;
> struct scsi_device* d;
>
> #if (LINUX_VERSION_CODE >= KERNEL_VERSION(2,5,65))
> # if ((LINUX_VERSION_CODE > KERNEL_VERSION(2,6,0)) ||
> defined(shost_for_each_device))
> + spin_unlock(pHba->host->host_lock);

Mark,
your patch helps, however I would note that it is fully incorrect: scsi host
reset handler on kernels > KERNEL_VERSION(2,6,12) do not take host_lock.

Also I've found yet another issue: your driver is not frees allocted resources,
if it loaded after i2o_block driver. Please see patch in attachments.

Thank you,
 Vasily Averin

SWsoft Virtuozzo/OpenVZ Linux kernel team

> shost_for_each_device(d, pHba->host) {
> # else
> list_for_each_entry(d, &pHba->host->my_devices, siblings) {
> # endif
> unsigned long flags;
> spin_lock_irqsave(&d->list_lock, flags);
> list_for_each_entry(cmd, &d->cmd_list, list) {
> if (cmd->serial_number == 0) {
> continue;
> }
> cmd->result = (DID_OK << 16) | (QUEUE_FULL <<
> 1);
> cmd->scsi_done(cmd);
> }

>   spin_unlock_irqrestore(&d->list_lock, flags);
>   }
> +# if ((LINUX_VERSION_CODE > KERNEL_VERSION(2,6,0)) ||
> defined(shost_for_each_device))
> +   spin_lock(pHba->host->host_lock);
> +# endif
> #else
>   d = pHba->host->host_queue;
>
> Sincerely -- Mark Salyzyn
>
>
>>-----Original Message-----
>>From: Vasily Averin [mailto:vvs@sw.ru]
>>Sent: Monday, August 14, 2006 10:02 AM
>>To: Salyzyn, Mark
>>Cc: Markus Lidel; devel@openvz.org
>>Subject: Re: i2o hardware hangs (ASR-2010S)
>>
>>
>>Hello Mark,
>>
>>I've tested your driver and unfortunately found bug in scsi
>>host reset handler:
>>
>>adpt_reset (on kernels <= KERNEL_VERSION(2,6,12) it called
>>with host_lock taken)
>> adpt_hba_reset
>> adpt_fail_posted_scbs
>>   shost_for_each_device
>>   __scsi_iterate_devices
>>   spin_lock_irqsave(shost->host_lock, flags); <<<<< deadlock
>>
>>Also I've noticed that adpt_hba_reset() can be called also
>>from adpt_ioctl() and
>>it have taken host_lock too on the kernel >= KERNEL_VERSION(2,5,65).
>>
>>However currently I do not understand how to fix this issue correctly.
>>
>>Thank you,
>> Vasily Averin
>>
>>Salyzyn, Mark wrote:
>>>I had sent you the driver source in a previous email, I am
>>sending it
>>>again. Please keep me in the loop since latest model
>>kernels (we have
>>>customers that confirm 2.6.16) may require changes in the driver to

>>>compile.
>>>
>>>Since the kernel.org policy is to focus on the i2o driver
>>being beefed
>>>up, no patches or changes are accepted for the dpt_i2o
>>driver into the
>>>kernel. Sad that we had just finished a stint beefing up the dpt_i2o
>>>driver just before that decision was made ...
>>>
>>>The comments about error recovery were meant as a starting point, it
>>>looks like Markus will have the final say.
>>>
>>>As for the timeouts, I referred to DASD (Disk) targets. 3 minute for
>>>RAID devices in a rolling timeout is used to deal with
>>situations that
>>>require a complete spin up of all component drives, or to deal with
>>>worst case error recovery scenarios. Individual DASD targets, on the
>>>other hand, should report back within 30 seconds for I/O. None DASD
>>>targets are all direct, and thus should respect any
>>timeouts set by the
>>>system (if any).
>>>
>>>Sincerely -- Mark Salyzyn
>>>
>>>>-----Original Message-----
>>>>From: Vasily Averin [mailto:vvs@sw.ru]
>>>>Sent: Tuesday, August 08, 2006 5:48 AM
>>>>To: Salyzyn, Mark
>>>>Cc: Markus Lidel; devel@openvz.org
>>>>Subject: Re: i2o hardware hangs (ASR-2010S)
>>>>
>>>>
>>>>Mark,
>>>>
>>>>Salyzyn, Mark wrote:
>>>>>Vasily, it will necessarily be up to you as to whether you
>>switch to
>>>>>dpt_i2o to get the hardening you require today, or work out
>>>>a deal with
>>>>>Markus to add timeout/reset functionality to the i2o driver.
>>>>Of course, you are right. Currently our customers have bad 2
>>>>alternatives:
>>>>- be tolerate to these hangs
>>>>- if they can't bear it -- replace i2o hardware
>>>>
>>>>Therefore first at all I'm going to add third possible
>>>>alternative, dpt_i2o driver.
>>>>

>>>>Mark, could you please send me latest version of your driver
>>>>directly? Or can I
>>>>probably take it from mainstream?
>>>>
>>>>The next task is help Markus in i2o error/reset handler
>>>>implementation.
>>>>
>>>>>My recommendations for the i2o driver reset procedure is to use a
>>>>>rolling timeout, every new command completion resets the
>>>>global timer.
>>>>>This will allow starved or long commands to process. Once
>>>>the timer hits
>>>>>3 minutes for RAID (Block or SCSI) targets that have multiple
>>>>>inheritances, 30 seconds for SCSI DASD targets, or some
>>>>insmod tunable,
>>>>>it resets the adapter. I recommend that when we hit ten
>>>>seconds, or some
>>>>>insmod tunable, that we call a card specific health check
>>>>routine. I do
>>>>>not recommend health check polling because we have noticed
>>>>a reduction
>>>>>in Adapter performance in some systems and generic i2o cards would
>>>>>require a command to check, so that is why I tie it to the
>>>>ten seconds
>>>>>past last completion. For the DPT/Adaptec series of
>>>>adapters, it checks
>>>>>the BlinkLED status (code fragment in dpt_i2o driver at
>>>>>adpt_read_blink_led), and if set, immediately record the
>>>>fact and resets
>>>>>the adapter. For cards other than the DPT/Adaptec series, I
>>>>recommend a
>>>>>short timeout Get Status request to see if the Firmware is in a run
>>>>>state and is responsive to this simple command. The reset
>>>>code will need
>>>>>to retry all commands itself, I do not believe the block
>>>>system has an
>>>>>error status that can be used for it to retry the commands.
>>>>If the Reset
>>>>>Iop in the reset adapter code is unresponsive, then the
>>>>known targets
>>>>>need to be placed offline.
>>>>Sorry, I do not have your big experience in scsi and do not
>>>>know nothing in i2o.
>>>>However are you sure than 3 min is enough for timeout? As far
>>>>as I know some
>>>>scsi commands (for example rewind on tapes) can last during a
>>>>very long time.
>>>>

>>>>Also I have some other questions but currently I'm not fell
>>>>that I'm ready for
>>>>this discussion.
>>>>
>>>>Thank you,
>>>> Vasily Averin
>>>>
>>>>SWsoft Virtuozzo/OpenVZ Linux kernel team

```
--- ./dpt_i2o.c.d2o2 2006-08-16 06:21:25.000000000 +0400
+++ ./dpt_i2o.c 2006-08-16 06:22:16.000000000 +0400
@@ -4532,17 +4532,17 @@ static int __init dpt_init(void)
 #endif

  error = pci_register_driver(&dpt_pci_driver);
-#if ((LINUX_VERSION_CODE < KERNEL_VERSION(2,5,0)) &&
defined(SCSI_HAS_SCSI_IN_DETECTION))
+
  if (error < 0 || hba_count == 0) {
+  pci_unregister_driver(&dpt_pci_driver);
+#if ((LINUX_VERSION_CODE < KERNEL_VERSION(2,5,0)) &&
defined(SCSI_HAS_SCSI_IN_DETECTION))
    scsi_unregister_module(MODULE_SCSI_HA,&driver_template);
+#endif
+#ifdef REBOOT_NOTIFIER
+  unregister_reboot_notifier(&adpt_reboot_notifier);
+#endif
   return (error < 0) ? error : -ENODEV;
  }
-#else
- if (error < 0)
-  return error;
- if (hba_count == 0)
-  return -ENODEV;
-#endif
  /* In INIT state, Activate IOPs */
  for (pHba = hba_chain; pHba; pHba = pHba->next) {
   // Activate does get status , init outbound, and get hrt
```