Subject: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Fri, 04 Aug 2006 11:49:30 GMT
View Forum Message <> Reply to Message

Hello Markus,

We experience problems with I2O hardware on 2.6 kernels, probably this can help you or maybe you even know the answer. Can you please, take a look?

After migration to 2.6 kernels our customers began to claim that i2o-based nodes hang. We have investigated these claims and discovered that i2o disks on theses nodes stopped the processing of any IO requests. Please, note, it is not a single issue, it happens from time to time.

Our kernel-space watchdog module has produced the following output to serial console

Jul 31 07:38:37
(80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626 1034743472 407332291)
Jul 31 07:39:38
(80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364 1034764548 407338084)
(80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375 1034808048 407351490)
(80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317 1034845588 407364374)
(80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031 1034865212 407369386)
(80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676 1034892406 407377617)
(80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294 1034921708 407385382)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928376 407387163)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928378 407387163)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928384 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928384 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928384 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928386 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928390 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928390 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928390 407387164)
(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966 1034928390 407387164)

where r(reads, read_sectors, read_merges) w(writes, write_sectors, write_merges)

Magic keys works, according to showProcess processors are in idle, ShowTraces shows a few thousand processes in D-state, but we can not find any deadlocks, it looks like the processes waits until I/O finished. Unfortunately i2o layer has no any error handlers and there is no any chance that the node will return from this coma.

Described incident has occurred after ~2 weeks uptime. It was Supermicro X5DP8 motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel 2.6.8-022stab078.9-enterprise, sources/configs are accessible on openvz.org.

In the bootlogs I've found mtrr message. As far as I know you have fixed this issue, however I'm not sure that it can leads to described hang.

```
I2O Core - (C) Copyright 1999 Red Hat Software
i2o: max_drivers=4
i2o: Checking for PCI I2O controllers...
ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
i2o: I2O controller found on bus 6 at 8.
i2o: PCI I2O controller
     BAR0 at 0xF8400000 size=1048576
     BAR1 at 0xFB000000 size=16777216
mtrr: type mismatch for fb000000,1000000 old: uncachable new: write-combining
i2o: could not enable write combining MTRR
iop0: Installed at IRQ 72
iop0: Activating I2O controller...
iop0: This may take a few minutes if there are many devices
iop0: HRT has 1 entries of 16 bytes each.
Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
iop0: Controller added
I2O Block Storage OSM v0.9
   (c) Copyright 1999-2001 Red Hat Software.
block-osm: registered device at major 80
block-osm: New device detected (TID: 211)
Using anticipatory io scheduler
 i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >

# cat /proc/mtrr
reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
reg01: base=0x00000000 (   0MB), size=8192MB: write-back, count=1
reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
```

I would repeat, it is not a single fault, we have received similar claims once and again. For some time we believed that it was due some hardware faults, however some doubts are cast upon it. The same nodes worked well long time ago without any troubles under 2.4-based kernels with dpt_i2o driver and we have not observed any of i2o hardware troubles so frequently.

Is it possible that our kernel (based on 2.6.8.1 mainstream) have some bugs in i2o drivers? However we're using driver sources taken from RHEL4U2 kernel, and I cannot find any similar claims from RHEL4 customers.

Is it possible than we have some other related kernels bugs? In this case why we have such kind of issues only on i2o-based nodes?

Could you please give me some hints which allow me to continue investigation of this issue. If you have any suggestions I'll check them next time.

Thank you,
Vasily Averin

SWsoft Virtuozzo/OpenVZ Linux kernel team

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by mark_salyzyn on Fri, 04 Aug 2006 12:06:52 GMT
View Forum Message <> Reply to Message

Markus, when the commands time out, do you perform a reset iop sequence?
I thought you added the BlinkLED code detection that is in the dpt_i2o
driver, if not, we should make sure it is there so that we get a report
in the console and an accompanying reset. Vasily, you console log did
not report anything at the time of failure, I would have expected some
timeout reports.

If it will help, Vasily, contact me for the latest dpt_i2o driver as
that is the driver I am most familiar with; it may be of interest to
determine if the problem duplicates with the dpt_i2o driver. Keep in
mind that the i2o driver is a block driver, dpt_i2o is a scsi driver.

Sincerely -- Mark Salyzyn

> -----Original Message-----
> From: linux-scsi-owner@vger.kernel.org
> [mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of Vasily Averin
> Sent: Friday, August 04, 2006 7:50 AM
> To: linux-scsi@vger.kernel.org; Markus Lidel
> Cc: devel@openvz.org
> Subject: i2o hardware hangs (ASR-2010S)
>
>
> Hello Markus,
>
> We experience problems with I2O hardware on 2.6 kernels,
> probably this can help
> you or maybe you even know the answer. Can you please, take a look?
>
> After migration to 2.6 kernels our customers began to claim
> that i2o-based
> nodes hang. We have investigated these claims and discovered
> that i2o disks on
> theses nodes stopped the processing of any IO requests.
> Please, note, it is not
> a single issue, it happens from time to time.
>
> Our kernel-space watchdog module has produced the following

> output to serial console
>
> Jul 31 07:38:37
> (80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
> 1034743472 407332291)
> Jul 31 07:39:38
> (80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
> 1034764548 407338084)
> (80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
> 1034808048 407351490)
> (80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
> 1034845588 407364374)
> (80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
> 1034865212 407369386)
> (80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
> 1034892406 407377617)
> (80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
> 1034921708 407385382)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928376 407387163)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928378 407387163)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928384 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928384 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928384 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928386 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928390 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928390 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928390 407387164)
> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> 1034928390 407387164)
>
> where r(reads, read_sectors, read_merges) w(writes,
> write_sectors, write_merges)
>
> Magic keys works, according to showProcess processors are in
> idle, ShowTraces
> shows a few thousand processes in D-state, but we can not
> find any deadlocks, it
> looks like the processes waits until I/O finished.
> Unfortunately i2o layer has

> no any error handlers and there is no any chance that the
> node will return
> from this coma.
>
> Described incident has occurred after ~2 weeks uptime. It was
> Supermicro X5DP8
> motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
> 2.6.8-022stab078.9-enterprise, sources/configs are accessible
> on openvz.org.
>
> In the bootlogs I've found mtrr message. As far as I know you
> have fixed this
> issue, however I'm not sure that it can leads to described hang.
>
> I2O Core - (C) Copyright 1999 Red Hat Software
> i2o: max_drivers=4
> i2o: Checking for PCI I2O controllers...
> ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
> i2o: I2O controller found on bus 6 at 8.
> i2o: PCI I2O controller
>   BAR0 at 0xF8400000 size=1048576
>   BAR1 at 0xFB000000 size=16777216
> mtrr: type mismatch for fb000000,1000000 old: uncachable new:
> write-combining
> i2o: could not enable write combining MTRR
> iop0: Installed at IRQ 72
> iop0: Activating I2O controller...
> iop0: This may take a few minutes if there are many devices
> iop0: HRT has 1 entries of 16 bytes each.
> Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
> iop0: Controller added
> I2O Block Storage OSM v0.9
>   (c) Copyright 1999-2001 Red Hat Software.
> block-osm: registered device at major 80
> block-osm: New device detected (TID: 211)
> Using anticipatory io scheduler
> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
>
> # cat /proc/mtrr
> reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
> reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
> reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
> reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
>
> I would repeat, it is not a single fault, we have received
> similar claims once
> and again. For some time we believed that it was due some
> hardware faults,

> however some doubts are cast upon it. The same nodes worked
> well long time ago
> without any troubles under 2.4-based kernels with dpt_i2o
> driver and we have not
> observed any of i2o hardware troubles so frequently.
>
> Is it possible that our kernel (based on 2.6.8.1 mainstream)
> have some bugs in
> i2o drivers? However we're using driver sources taken from
> RHEL4U2 kernel, and I
> cannot find any similar claims from RHEL4 customers.
>
> Is it possible than we have some other related kernels bugs?
> In this case why we
> have such kind of issues only on i2o-based nodes?
>
> Could you please give me some hints which allow me to
> continue investigation of
> this issue. If you have any suggestions I'll check them next time.
>
> Thank you,
> Vasily Averin
>
> SWsoft Virtuozzo/OpenVZ Linux kernel team
>
> -
> To unsubscribe from this list: send the line "unsubscribe
> linux-scsi" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
>

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Mon, 07 Aug 2006 08:04:40 GMT
View Forum Message <> Reply to Message

Hello Mark,

thank you for your assistance.

Salyzyn, Mark wrote:
> Markus, when the commands time out, do you perform a reset iop sequence?
> I thought you added the BlinkLED code detection that is in the dpt_i2o
> driver, if not, we should make sure it is there so that we get a report
> in the console and an accompanying reset. Vasily, you console log did
> not report anything at the time of failure, I would have expected some
> timeout reports.

Unfortunately console logs does not have any errors or timeout reports.
If you wish, I can send you console logs directly.

However as far as I understand i2o layer does not have any sort of timeout/error handlers (I hope Markus correct me if I'm err), and it would be great if this feature will be appear in the future.

> If it will help, Vasily, contact me for the latest dpt_i2o driver as
> that is the driver I am most familiar with; it may be of interest to
> determine if the problem duplicates with the dpt_i2o driver. Keep in
> mind that the i2o driver is a block driver, dpt_i2o is a scsi driver.

Unfortunately we do not know how we can reproduce this issue. Currently it occurs on the production nodes only and customers are very against of any experiments on these nodes.

Therefore it is not to easy to switch from i2o layer to your dpt_i2o driver.

Currently we have not dpt_i2o driver in our kernels. The most important reasons are:
- this driver did have some problems on 64-bit kernels (but it is resolved already, I'm I right?).
- it is not included into 2.6-based Red Hat distributiuons.
- it did not worked when I've tried to compile it into kernel.
- when I've tried to build it as module, I've discovered that it conflicts with i2o drivers: initscripts on the some distributions (FC4?) have tried to load both of these modules (one from initrd, second -- when detects according PCIID) and it hangs the node. I've not found any working combination and therefore we've decided to not include dpt_i2o driver into our 2.6 kernels.

However, Mark, I'm ready to check your new driver on our internal testnodes, and if last issue (modules conflicts) is fixed I'll try to include your driver into our kernels.

Thank you,
Vasily Averin

> Sincerely -- Mark Salyzyn
>
>>-----Original Message-----
>>From: linux-scsi-owner@vger.kernel.org
>>[mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of Vasily Averin
>>Sent: Friday, August 04, 2006 7:50 AM
>>To: linux-scsi@vger.kernel.org; Markus Lidel
>>Cc: devel@openvz.org
>>Subject: i2o hardware hangs (ASR-2010S)
>>
>>

>>Hello Markus,
>>
>>We experience problems with I2O hardware on 2.6 kernels,
>>probably this can help
>>you or maybe you even know the answer. Can you please, take a look?
>>
>>After migration to 2.6 kernels our customers began to claim
>>that i2o-based
>>nodes hang. We have investigated these claims and discovered
>>that i2o disks on
>>theses nodes stopped the processing of any IO requests.
>>Please, note, it is not
>>a single issue, it happens from time to time.
>>
>>Our kernel-space watchdog module has produced the following
>>output to serial console
>>
>>Jul 31 07:38:37
>>(80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
>>1034743472 407332291)
>>Jul 31 07:39:38
>>(80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
>>1034764548 407338084)
>>(80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
>>1034808048 407351490)
>>(80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
>>1034845588 407364374)
>>(80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
>>1034865212 407369386)
>>(80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
>>1034892406 407377617)
>>(80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
>>1034921708 407385382)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928376 407387163)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928378 407387163)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928384 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928384 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928384 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928386 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928390 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966

>>1034928390 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928390 407387164)
>>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>1034928390 407387164)
>>
>>where r(reads, read_sectors, read_merges) w(writes,
>>write_sectors, write_merges)
>>
>>Magic keys works, according to showProcess processors are in
>>idle, ShowTraces
>>shows a few thousand processes in D-state, but we can not
>>find any deadlocks, it
>>looks like the processes waits until I/O finished.
>>Unfortunately i2o layer has
>>no any error handlers and there is no any chance that the
>>node will return
>>from this coma.
>>
>>Described incident has occurred after ~2 weeks uptime. It was
>>Supermicro X5DP8
>>motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
>>2.6.8-022stab078.9-enterprise, sources/configs are accessible
>>on openvz.org.
>>
>>In the bootlogs I've found mtrr message. As far as I know you
>>have fixed this
>>issue, however I'm not sure that it can leads to described hang.
>>
>>I2O Core - (C) Copyright 1999 Red Hat Software
>>i2o: max_drivers=4
>>i2o: Checking for PCI I2O controllers...
>>ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
>>i2o: I2O controller found on bus 6 at 8.
>>i2o: PCI I2O controller
>>   BAR0 at 0xF8400000 size=1048576
>>   BAR1 at 0xFB000000 size=16777216
>>mtrr: type mismatch for fb000000,1000000 old: uncachable new:
>>write-combining
>>i2o: could not enable write combining MTRR
>>iop0: Installed at IRQ 72
>>iop0: Activating I2O controller...
>>iop0: This may take a few minutes if there are many devices
>>iop0: HRT has 1 entries of 16 bytes each.
>>Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
>>iop0: Controller added
>>I2O Block Storage OSM v0.9
>>   (c) Copyright 1999-2001 Red Hat Software.

>>block-osm: registered device at major 80
>>block-osm: New device detected (TID: 211)
>>Using anticipatory io scheduler
>> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
>>
>># cat /proc/mtrr
>>reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
>>reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
>>reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
>>reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
>>
>>I would repeat, it is not a single fault, we have received
>>similar claims once
>>and again. For some time we believed that it was due some
>>hardware faults,
>>however some doubts are cast upon it. The same nodes worked
>>well long time ago
>>without any troubles under 2.4-based kernels with dpt_i2o
>>driver and we have not
>>observed any of i2o hardware troubles so frequently.
>>
>>Is it possible that our kernel (based on 2.6.8.1 mainstream)
>>have some bugs in
>>i2o drivers? However we're using driver sources taken from
>>RHEL4U2 kernel, and I
>>cannot find any similar claims from RHEL4 customers.
>>
>>Is it possible than we have some other related kernels bugs?
>>In this case why we
>>have such kind of issues only on i2o-based nodes?
>>
>>Could you please give me some hints which allow me to
>>continue investigation of
>>this issue. If you have any suggestions I'll check them next time.
>>
>>Thank you,
>>   Vasily Averin
>>
>>SWsoft Virtuozzo/OpenVZ Linux kernel team
>>
>>-
>>To unsubscribe from this list: send the line "unsubscribe
>>linux-scsi" in
>>the body of a message to majordomo@vger.kernel.org
>>More majordomo info at http://vger.kernel.org/majordomo-info.html
>>
>
>

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by Markus Lidel on Mon, 07 Aug 2006 09:23:16 GMT
View Forum Message <> Reply to Message

Hello,

Salyzyn, Mark wrote:
> Markus, when the commands time out, do you perform a reset iop sequence?

Currently not.

> I thought you added the BlinkLED code detection that is in the dpt_i2o
> driver, if not, we should make sure it is there so that we get a report
> in the console and an accompanying reset. Vasily, you console log did
> not report anything at the time of failure, I would have expected some
> timeout reports.

I agree, there should be some notice that a command timed out.

[...]

>> Described incident has occurred after ~2 weeks uptime. It was
>> Supermicro X5DP8
>> motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
>> 2.6.8-022stab078.9-enterprise, sources/configs are accessible
>> on openvz.org.

Could you please try out a current kernel?

Best regards,

Markus Lidel

Markus Lidel (Senior IT Consultant)

Shadow Connect GmbH
Carl-Reisch-Weg 12
D-86381 Krumbach
Germany

Phone: +49 82 82/99 51-0
Fax: +49 82 82/99 51-11

E-Mail: Markus.Lidel@shadowconnect.com
URL: http://www.shadowconnect.com

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by mark_salyzyn on Mon, 07 Aug 2006 11:44:41 GMT
View Forum Message <> Reply to Message

64 bit (address and datapath) works in the driver I have provided,
although we have heard of some SM motherboards that work with these ZCR
cards that have broken bridges. The interference issue required both
drivers to register the address range, the sources I have provided
perform the registration, you may have to check with Markus to see if
the version of the i2o driver utilizes the same.

It was decided by the community to deprecate the dpt_i2o driver in the
2.6 kernel, it still remains but any bugfixes are rejected unless they
are minor. Adaptec is committed to supporting the dpt_i2o driver for OEM
customers. Markus has taken efforts to incorporate the dpt_i2o features,
64 bit etc, in the i2o driver. I do hope he has incorporated a timeout
and recovery mechanism, it is not dpt_i2o specific. I look forward to
his comments.

Sincerely -- Mark Salyzyn

> -----Original Message-----
> From: Vasily Averin [mailto:vvs@sw.ru]
> Sent: Monday, August 07, 2006 4:05 AM
> To: Salyzyn, Mark
> Cc: Markus Lidel; devel@openvz.org
> Subject: Re: i2o hardware hangs (ASR-2010S)
>
>
> Hello Mark,
>
> thank you for your assistance.
>
> Salyzyn, Mark wrote:
> > Markus, when the commands time out, do you perform a reset
> iop sequence?
> > I thought you added the BlinkLED code detection that is in
> the dpt_i2o
> > driver, if not, we should make sure it is there so that we
> get a report
> > in the console and an accompanying reset. Vasily, you
> console log did
> > not report anything at the time of failure, I would have
> expected some
> > timeout reports.
>
> Unfortunately console logs does not have any errors or
> timeout reports.
> If you wish, I can send you console logs directly.

>
> However as far as I understand i2o layer does not have any
> sort of timeout/error
> handlers (I hope Markus correct me if I'm err), and it would
> be great if this
> feature will be appear in the future.
>
> > If it will help, Vasily, contact me for the latest dpt_i2o driver as
> > that is the driver I am most familiar with; it may be of interest to
> > determine if the problem duplicates with the dpt_i2o driver. Keep in
> > mind that the i2o driver is a block driver, dpt_i2o is a
> scsi driver.
>
> Unfortunately we do not know how we can reproduce this issue.
> Currently it
> occurs on the production nodes only and customers are very
> against of any
> experiments on these nodes.
>
> Therefore it is not to easy to switch from i2o layer to your
> dpt_i2o driver.
>
> Currently we have not dpt_i2o driver in our kernels. The most
> important reasons are:
> - this driver did have some problems on 64-bit kernels (but
> it is resolved
> already, I'm I right?).
> - it is not included into 2.6-based Red Hat distributiuons.
> - it did not worked when I've tried to compile it into kernel.
> - when I've tried to build it as module, I've discovered that
> it conflicts with
> i2o drivers: initscripts on the some distributions (FC4?)
> have tried to load
> both of these modules (one from initrd, second -- when
> detects according PCIID)
> and it hangs the node. I've not found any working combination
> and therefore
> we've decided to not include dpt_i2o driver into our 2.6 kernels.
>
> However, Mark, I'm ready to check your new driver on our
> internal testnodes, and
> if last issue (modules conflicts) is fixed I'll try to
> include your driver into
> our kernels.
>
> Thank you,
> Vasily Averin
>

> > Sincerely -- Mark Salyzyn
> >
> >>-----Original Message-----
> >>From: linux-scsi-owner@vger.kernel.org
> >>[mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of Vasily Averin
> >>Sent: Friday, August 04, 2006 7:50 AM
> >>To: linux-scsi@vger.kernel.org; Markus Lidel
> >>Cc: devel@openvz.org
> >>Subject: i2o hardware hangs (ASR-2010S)
> >>
> >>
> >>Hello Markus,
> >>
> >>We experience problems with I2O hardware on 2.6 kernels,
> >>probably this can help
> >>you or maybe you even know the answer. Can you please, take a look?
> >>
> >>After migration to 2.6 kernels our customers began to claim
> >>that i2o-based
> >>nodes hang. We have investigated these claims and discovered
> >>that i2o disks on
> >>theses nodes stopped the processing of any IO requests.
> >>Please, note, it is not
> >>a single issue, it happens from time to time.
> >>
> >>Our kernel-space watchdog module has produced the following
> >>output to serial console
> >>
> >>Jul 31 07:38:37
> >>(80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
> >>1034743472 407332291)
> >>Jul 31 07:39:38
> >>(80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
> >>1034764548 407338084)
> >>(80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
> >>1034808048 407351490)
> >>(80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
> >>1034845588 407364374)
> >>(80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
> >>1034865212 407369386)
> >>(80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
> >>1034892406 407377617)
> >>(80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
> >>1034921708 407385382)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928376 407387163)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928378 407387163)

> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928384 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928384 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928384 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928386 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928390 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928390 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928390 407387164)
> >>(80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>1034928390 407387164)
> >>
> >>where r(reads, read_sectors, read_merges) w(writes,
> >>write_sectors, write_merges)
> >>
> >>Magic keys works, according to showProcess processors are in
> >>idle, ShowTraces
> >>shows a few thousand processes in D-state, but we can not
> >>find any deadlocks, it
> >>looks like the processes waits until I/O finished.
> >>Unfortunately i2o layer has
> >>no any error handlers and there is no any chance that the
> >>node will return
> >>from this coma.
> >>
> >>Described incident has occurred after ~2 weeks uptime. It was
> >>Supermicro X5DP8
> >>motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
> >>2.6.8-022stab078.9-enterprise, sources/configs are accessible
> >>on openvz.org.
> >>
> >>In the bootlogs I've found mtrr message. As far as I know you
> >>have fixed this
> >>issue, however I'm not sure that it can leads to described hang.
> >>
> >>I2O Core - (C) Copyright 1999 Red Hat Software
> >>i2o: max_drivers=4
> >>i2o: Checking for PCI I2O controllers...
> >>ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
> >>i2o: I2O controller found on bus 6 at 8.
> >>i2o: PCI I2O controller
> >> BAR0 at 0xF8400000 size=1048576
> >> BAR1 at 0xFB000000 size=16777216

> >>mtrr: type mismatch for fb000000,1000000 old: uncachable new:
> >>write-combining
> >>i2o: could not enable write combining MTRR
> >>iop0: Installed at IRQ 72
> >>iop0: Activating I2O controller...
> >>iop0: This may take a few minutes if there are many devices
> >>iop0: HRT has 1 entries of 16 bytes each.
> >>Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
> >>iop0: Controller added
> >>I2O Block Storage OSM v0.9
> >> (c) Copyright 1999-2001 Red Hat Software.
> >>block-osm: registered device at major 80
> >>block-osm: New device detected (TID: 211)
> >>Using anticipatory io scheduler
> >> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
> >>
> >># cat /proc/mtrr
> >>reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
> >>reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
> >>reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
> >>reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
> >>
> >>I would repeat, it is not a single fault, we have received
> >>similar claims once
> >>and again. For some time we believed that it was due some
> >>hardware faults,
> >>however some doubts are cast upon it. The same nodes worked
> >>well long time ago
> >>without any troubles under 2.4-based kernels with dpt_i2o
> >>driver and we have not
> >>observed any of i2o hardware troubles so frequently.
> >>
> >>Is it possible that our kernel (based on 2.6.8.1 mainstream)
> >>have some bugs in
> >>i2o drivers? However we're using driver sources taken from
> >>RHEL4U2 kernel, and I
> >>cannot find any similar claims from RHEL4 customers.
> >>
> >>Is it possible than we have some other related kernels bugs?
> >>In this case why we
> >>have such kind of issues only on i2o-based nodes?
> >>
> >>Could you please give me some hints which allow me to
> >>continue investigation of
> >>this issue. If you have any suggestions I'll check them next time.
> >>
> >>Thank you,
> >> Vasily Averin

> >>
> >>SWsoft Virtuozzo/OpenVZ Linux kernel team
> >>
> >>-
> >>To unsubscribe from this list: send the line "unsubscribe
> >>linux-scsi" in
> >>the body of a message to majordomo@vger.kernel.org
> >>More majordomo info at http://vger.kernel.org/majordomo-info.html
> >>
> >
> >
>
>
>

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by Markus Lidel on Mon, 07 Aug 2006 14:33:20 GMT
View Forum Message <> Reply to Message

Hello,

Salyzyn, Mark wrote:
> 64 bit (address and datapath) works in the driver I have provided,
> although we have heard of some SM motherboards that work with these ZCR
> cards that have broken bridges. The interference issue required both
> drivers to register the address range, the sources I have provided
> perform the registration, you may have to check with Markus to see if
> the version of the i2o driver utilizes the same.
> It was decided by the community to deprecate the dpt_i2o driver in the
> 2.6 kernel, it still remains but any bugfixes are rejected unless they
> are minor. Adaptec is committed to supporting the dpt_i2o driver for OEM
> customers. Markus has taken efforts to incorporate the dpt_i2o features,
> 64 bit etc, in the i2o driver. I do hope he has incorporated a timeout
> and recovery mechanism, it is not dpt_i2o specific. I look forward to
> his comments.

At the moment there is no recovery mechanism in case of a timeout in the
I2O driver. I think it could be a little bit problematic to reset the
controller in case a timeout occur, because all open operations are lost
in this case. But i agree that at least an error message should be logged
to inform the user something is going wrong.

>> -----Original Message-----
>> From: Vasily Averin [mailto:vvs@sw.ru]
>> Sent: Monday, August 07, 2006 4:05 AM
>> To: Salyzyn, Mark
>> Cc: Markus Lidel; devel@openvz.org

>> Subject: Re: i2o hardware hangs (ASR-2010S)
>>
>>
>> Hello Mark,
>>
>> thank you for your assistance.
>>
>> Salyzyn, Mark wrote:
>>> Markus, when the commands time out, do you perform a reset
>> iop sequence?
>>> I thought you added the BlinkLED code detection that is in
>> the dpt_i2o
>>> driver, if not, we should make sure it is there so that we
>> get a report
>>> in the console and an accompanying reset. Vasily, you
>> console log did
>>> not report anything at the time of failure, I would have
>> expected some
>>> timeout reports.
>> Unfortunately console logs does not have any errors or
>> timeout reports.
>> If you wish, I can send you console logs directly.
>>
>> However as far as I understand i2o layer does not have any
>> sort of timeout/error
>> handlers (I hope Markus correct me if I'm err), and it would
>> be great if this
>> feature will be appear in the future.
>>
>>> If it will help, Vasily, contact me for the latest dpt_i2o driver as
>>> that is the driver I am most familiar with; it may be of interest to
>>> determine if the problem duplicates with the dpt_i2o driver. Keep in
>>> mind that the i2o driver is a block driver, dpt_i2o is a
>> scsi driver.
>>
>> Unfortunately we do not know how we can reproduce this issue.
>> Currently it
>> occurs on the production nodes only and customers are very
>> against of any
>> experiments on these nodes.
>>
>> Therefore it is not to easy to switch from i2o layer to your
>> dpt_i2o driver.
>>
>> Currently we have not dpt_i2o driver in our kernels. The most
>> important reasons are:
>> - this driver did have some problems on 64-bit kernels (but
>> it is resolved

>> already, I'm I right?).
>> - it is not included into 2.6-based Red Hat distributiuons.
>> - it did not worked when I've tried to compile it into kernel.
>> - when I've tried to build it as module, I've discovered that
>> it conflicts with
>> i2o drivers: initscripts on the some distributions (FC4?)
>> have tried to load
>> both of these modules (one from initrd, second -- when
>> detects according PCIID)
>> and it hangs the node. I've not found any working combination
>> and therefore
>> we've decided to not include dpt_i2o driver into our 2.6 kernels.
>>
>> However, Mark, I'm ready to check your new driver on our
>> internal testnodes, and
>> if last issue (modules conflicts) is fixed I'll try to
>> include your driver into
>> our kernels.
>>
>> Thank you,
>> Vasily Averin
>>
>>> Sincerely -- Mark Salyzyn
>>>
>>>> -----Original Message-----
>>>> From: linux-scsi-owner@vger.kernel.org
>>>> [mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of Vasily Averin
>>>> Sent: Friday, August 04, 2006 7:50 AM
>>>> To: linux-scsi@vger.kernel.org; Markus Lidel
>>>> Cc: devel@openvz.org
>>>> Subject: i2o hardware hangs (ASR-2010S)
>>>>
>>>>
>>>> Hello Markus,
>>>>
>>>> We experience problems with I2O hardware on 2.6 kernels,
>>>> probably this can help
>>>> you or maybe you even know the answer. Can you please, take a look?
>>>>
>>>> After migration to 2.6 kernels our customers began to claim
>>>> that i2o-based
>>>> nodes hang. We have investigated these claims and discovered
>>>> that i2o disks on
>>>> theses nodes stopped the processing of any IO requests.
>>>> Please, note, it is not
>>>> a single issue, it happens from time to time.
>>>>
>>>> Our kernel-space watchdog module has produced the following

>>>> output to serial console
>>>>
>>>> Jul 31 07:38:37
>>>> (80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
>>>> 1034743472 407332291)
>>>> Jul 31 07:39:38
>>>> (80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
>>>> 1034764548 407338084)
>>>> (80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
>>>> 1034808048 407351490)
>>>> (80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
>>>> 1034845588 407364374)
>>>> (80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
>>>> 1034865212 407369386)
>>>> (80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
>>>> 1034892406 407377617)
>>>> (80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
>>>> 1034921708 407385382)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928376 407387163)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928378 407387163)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928384 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928384 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928384 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928386 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928390 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928390 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928390 407387164)
>>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>>> 1034928390 407387164)
>>>>
>>>> where r(reads, read_sectors, read_merges) w(writes,
>>>> write_sectors, write_merges)
>>>>
>>>> Magic keys works, according to showProcess processors are in
>>>> idle, ShowTraces
>>>> shows a few thousand processes in D-state, but we can not
>>>> find any deadlocks, it
>>>> looks like the processes waits until I/O finished.
>>>> Unfortunately i2o layer has

>>>> no any error handlers and there is no any chance that the
>>>> node will return
>>> >from this coma.
>>>> Described incident has occurred after ~2 weeks uptime. It was
>>>> Supermicro X5DP8
>>>> motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
>>>> 2.6.8-022stab078.9-enterprise, sources/configs are accessible
>>>> on openvz.org.
>>>>
>>>> In the bootlogs I've found mtrr message. As far as I know you
>>>> have fixed this
>>>> issue, however I'm not sure that it can leads to described hang.
>>>>
>>>> I2O Core - (C) Copyright 1999 Red Hat Software
>>>> i2o: max_drivers=4
>>>> i2o: Checking for PCI I2O controllers...
>>>> ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
>>>> i2o: I2O controller found on bus 6 at 8.
>>>> i2o: PCI I2O controller
>>>>   BAR0 at 0xF8400000 size=1048576
>>>>   BAR1 at 0xFB000000 size=16777216
>>>> mtrr: type mismatch for fb000000,1000000 old: uncachable new:
>>>> write-combining
>>>> i2o: could not enable write combining MTRR
>>>> iop0: Installed at IRQ 72
>>>> iop0: Activating I2O controller...
>>>> iop0: This may take a few minutes if there are many devices
>>>> iop0: HRT has 1 entries of 16 bytes each.
>>>> Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
>>>> iop0: Controller added
>>>> I2O Block Storage OSM v0.9
>>>>   (c) Copyright 1999-2001 Red Hat Software.
>>>> block-osm: registered device at major 80
>>>> block-osm: New device detected (TID: 211)
>>>> Using anticipatory io scheduler
>>>> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
>>>>
>>>> # cat /proc/mtrr
>>>> reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
>>>> reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
>>>> reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
>>>> reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
>>>>
>>>> I would repeat, it is not a single fault, we have received
>>>> similar claims once
>>>> and again. For some time we believed that it was due some
>>>> hardware faults,
>>>> however some doubts are cast upon it. The same nodes worked

>>>> well long time ago
>>>> without any troubles under 2.4-based kernels with dpt_i2o
>>>> driver and we have not
>>>> observed any of i2o hardware troubles so frequently.
>>>>
>>>> Is it possible that our kernel (based on 2.6.8.1 mainstream)
>>>> have some bugs in
>>>> i2o drivers? However we're using driver sources taken from
>>>> RHEL4U2 kernel, and I
>>>> cannot find any similar claims from RHEL4 customers.
>>>>
>>>> Is it possible than we have some other related kernels bugs?
>>>> In this case why we
>>>> have such kind of issues only on i2o-based nodes?
>>>>
>>>> Could you please give me some hints which allow me to
>>>> continue investigation of
>>>> this issue. If you have any suggestions I'll check them next time.
>>>>
>>>> Thank you,
>>>> Vasily Averin
>>>>
>>>> SWsoft Virtuozzo/OpenVZ Linux kernel team

Best regards,

Markus Lidel

Markus Lidel (Senior IT Consultant)

Shadow Connect GmbH
Carl-Reisch-Weg 12
D-86381 Krumbach
Germany

Phone: +49 82 82/99 51-0
Fax: +49 82 82/99 51-11

E-Mail: Markus.Lidel@shadowconnect.com
URL: http://www.shadowconnect.com

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by Markus Lidel on Mon, 07 Aug 2006 14:42:20 GMT
View Forum Message <> Reply to Message

Hello,

Vasily Averin wrote:
> thank you for your assistance.
> Salyzyn, Mark wrote:
>> Markus, when the commands time out, do you perform a reset iop sequence?
>> I thought you added the BlinkLED code detection that is in the dpt_i2o
>> driver, if not, we should make sure it is there so that we get a report
>> in the console and an accompanying reset. Vasily, you console log did
>> not report anything at the time of failure, I would have expected some
>> timeout reports.
> Unfortunately console logs does not have any errors or timeout reports.
> If you wish, I can send you console logs directly.
> However as far as I understand i2o layer does not have any sort of timeout/error
> handlers (I hope Markus correct me if I'm err), and it would be great if this
> feature will be appear in the future.

You're right, there is no timeout/error handling at all, because the I2O
spec says there must be a response to every message sent to the
controller. The controller must implement an timeout feature itself (for
example if a disk disappears during an operation, the controller take
care of it and informs the OS that something is wrong). But if this
really happens there should be at least a error message.

>> If it will help, Vasily, contact me for the latest dpt_i2o driver as
>> that is the driver I am most familiar with; it may be of interest to
>> determine if the problem duplicates with the dpt_i2o driver. Keep in
>> mind that the i2o driver is a block driver, dpt_i2o is a scsi driver.
>
> Unfortunately we do not know how we can reproduce this issue. Currently it
> occurs on the production nodes only and customers are very against of any
> experiments on these nodes.
> Therefore it is not to easy to switch from i2o layer to your dpt_i2o driver.
> Currently we have not dpt_i2o driver in our kernels. The most important reasons are:
> - this driver did have some problems on 64-bit kernels (but it is resolved
> already, I'm I right?).
> - it is not included into 2.6-based Red Hat distributiuons.
> - it did not worked when I've tried to compile it into kernel.
> - when I've tried to build it as module, I've discovered that it conflicts with
> i2o drivers: initscripts on the some distributions (FC4?) have tried to load
> both of these modules (one from initrd, second -- when detects according PCIID)
> and it hangs the node. I've not found any working combination and therefore
> we've decided to not include dpt_i2o driver into our 2.6 kernels.
> However, Mark, I'm ready to check your new driver on our internal testnodes, and
> if last issue (modules conflicts) is fixed I'll try to include your driver into
> our kernels.

With the latest version the module conflicts should be fixed.

>>> -----Original Message-----
>>> From: linux-scsi-owner@vger.kernel.org
>>> [mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of Vasily Averin
>>> Sent: Friday, August 04, 2006 7:50 AM
>>> To: linux-scsi@vger.kernel.org; Markus Lidel
>>> Cc: devel@openvz.org
>>> Subject: i2o hardware hangs (ASR-2010S)
>>>
>>>
>>> Hello Markus,
>>>
>>> We experience problems with I2O hardware on 2.6 kernels,
>>> probably this can help
>>> you or maybe you even know the answer. Can you please, take a look?
>>>
>>> After migration to 2.6 kernels our customers began to claim
>>> that i2o-based
>>> nodes hang. We have investigated these claims and discovered
>>> that i2o disks on
>>> theses nodes stopped the processing of any IO requests.
>>> Please, note, it is not
>>> a single issue, it happens from time to time.
>>>
>>> Our kernel-space watchdog module has produced the following
>>> output to serial console
>>>
>>> Jul 31 07:38:37
>>> (80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
>>> 1034743472 407332291)
>>> Jul 31 07:39:38
>>> (80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
>>> 1034764548 407338084)
>>> (80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
>>> 1034808048 407351490)
>>> (80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
>>> 1034845588 407364374)
>>> (80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
>>> 1034865212 407369386)
>>> (80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
>>> 1034892406 407377617)
>>> (80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
>>> 1034921708 407385382)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928376 407387163)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928378 407387163)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966

>>> 1034928384 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928384 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928384 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928386 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928390 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928390 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928390 407387164)
>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
>>> 1034928390 407387164)
>>>
>>> where r(reads, read_sectors, read_merges) w(writes,
>>> write_sectors, write_merges)
>>>
>>> Magic keys works, according to showProcess processors are in
>>> idle, ShowTraces
>>> shows a few thousand processes in D-state, but we can not
>>> find any deadlocks, it
>>> looks like the processes waits until I/O finished.
>>> Unfortunately i2o layer has
>>> no any error handlers and there is no any chance that the
>>> node will return
>> >from this coma.
>>> Described incident has occurred after ~2 weeks uptime. It was
>>> Supermicro X5DP8
>>> motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero Channel. Kernel
>>> 2.6.8-022stab078.9-enterprise, sources/configs are accessible
>>> on openvz.org.
>>>
>>> In the bootlogs I've found mtrr message. As far as I know you
>>> have fixed this
>>> issue, however I'm not sure that it can leads to described hang.
>>>
>>> I2O Core - (C) Copyright 1999 Red Hat Software
>>> i2o: max_drivers=4
>>> i2o: Checking for PCI I2O controllers...
>>> ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level, low) -> IRQ 72
>>> i2o: I2O controller found on bus 6 at 8.
>>> i2o: PCI I2O controller
>>>   BAR0 at 0xF8400000 size=1048576
>>>   BAR1 at 0xFB000000 size=16777216
>>> mtrr: type mismatch for fb000000,1000000 old: uncachable new:
>>> write-combining

>>> i2o: could not enable write combining MTRR
>>> iop0: Installed at IRQ 72
>>> iop0: Activating I2O controller...
>>> iop0: This may take a few minutes if there are many devices
>>> iop0: HRT has 1 entries of 16 bytes each.
>>> Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22 Function 0
>>> iop0: Controller added
>>> I2O Block Storage OSM v0.9
>>> (c) Copyright 1999-2001 Red Hat Software.
>>> block-osm: registered device at major 80
>>> block-osm: New device detected (TID: 211)
>>> Using anticipatory io scheduler
>>> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
>>>
>>> # cat /proc/mtrr
>>> reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
>>> reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
>>> reg02: base=0x200000000 (8192MB), size= 128MB: write-back, count=1
>>> reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
>>>
>>> I would repeat, it is not a single fault, we have received
>>> similar claims once
>>> and again. For some time we believed that it was due some
>>> hardware faults,
>>> however some doubts are cast upon it. The same nodes worked
>>> well long time ago
>>> without any troubles under 2.4-based kernels with dpt_i2o
>>> driver and we have not
>>> observed any of i2o hardware troubles so frequently.
>>>
>>> Is it possible that our kernel (based on 2.6.8.1 mainstream)
>>> have some bugs in
>>> i2o drivers? However we're using driver sources taken from
>>> RHEL4U2 kernel, and I
>>> cannot find any similar claims from RHEL4 customers.
>>>
>>> Is it possible than we have some other related kernels bugs?
>>> In this case why we
>>> have such kind of issues only on i2o-based nodes?
>>>
>>> Could you please give me some hints which allow me to
>>> continue investigation of
>>> this issue. If you have any suggestions I'll check them next time.
>>>
>>> Thank you,
>>> Vasily Averin
>>>
>>> SWsoft Virtuozzo/OpenVZ Linux kernel team

Best regards,

Markus Lidel

Markus Lidel (Senior IT Consultant)

Shadow Connect GmbH
Carl-Reisch-Weg 12
D-86381 Krumbach
Germany

Phone: +49 82 82/99 51-0
Fax: +49 82 82/99 51-11

E-Mail: Markus.Lidel@shadowconnect.com
URL: http://www.shadowconnect.com

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by mark_salyzyn on Mon, 07 Aug 2006 16:06:35 GMT
View Forum Message <> Reply to Message

The dpt_i2o driver has the advantage of all the requests tracked by the scsi subsystem, returning them as scsi queue full to be retried. Adpt_fail_posted_scbs is small miracle of simplicity. The i2o driver will have to maintain it's own queue of commands to add this functionality (!)

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. If you wish to attack this issue on your own and provide a patch to Markus, I am here to provide technical advice about the DPT/Adaptec I2O cards, but must admit a basic ignorance to Markus' driver sources and architecture.

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.

Sincerely -- Mark Salyzyn

> -----Original Message-----
> From: Markus Lidel [mailto:Markus.Lidel@shadowconnect.com]
> Sent: Monday, August 07, 2006 10:33 AM
> To: Salyzyn, Mark
> Cc: Vasily Averin; devel@openvz.org
> Subject: Re: i2o hardware hangs (ASR-2010S)
>
>
> Hello,
>
> Salyzyn, Mark wrote:
> > 64 bit (address and datapath) works in the driver I have provided,
> > although we have heard of some SM motherboards that work
> with these ZCR
> > cards that have broken bridges. The interference issue required both
> > drivers to register the address range, the sources I have provided
> > perform the registration, you may have to check with Markus
> to see if
> > the version of the i2o driver utilizes the same.
> > It was decided by the community to deprecate the dpt_i2o
> driver in the
> > 2.6 kernel, it still remains but any bugfixes are rejected
> unless they
> > are minor. Adaptec is committed to supporting the dpt_i2o
> driver for OEM
> > customers. Markus has taken efforts to incorporate the
> dpt_i2o features,
> > 64 bit etc, in the i2o driver. I do hope he has
> incorporated a timeout
> > and recovery mechanism, it is not dpt_i2o specific. I look
> forward to
> > his comments.
>
> At the moment there is no recovery mechanism in case of a
> timeout in the
> I2O driver. I think it could be a little bit problematic to reset the

> controller in case a timeout occur, because all open
> operations are lost
> in this case. But i agree that at least an error message
> should be logged
> to inform the user something is going wrong.
>
> >> -----Original Message-----
> >> From: Vasily Averin [mailto:vvs@sw.ru]
> >> Sent: Monday, August 07, 2006 4:05 AM
> >> To: Salyzyn, Mark
> >> Cc: Markus Lidel; devel@openvz.org
> >> Subject: Re: i2o hardware hangs (ASR-2010S)
> >>
> >>
> >> Hello Mark,
> >>
> >> thank you for your assistance.
> >>
> >> Salyzyn, Mark wrote:
> >>> Markus, when the commands time out, do you perform a reset
> >> iop sequence?
> >>> I thought you added the BlinkLED code detection that is in
> >> the dpt_i2o
> >>> driver, if not, we should make sure it is there so that we
> >> get a report
> >>> in the console and an accompanying reset. Vasily, you
> >> console log did
> >>> not report anything at the time of failure, I would have
> >> expected some
> >>> timeout reports.
> >> Unfortunately console logs does not have any errors or
> >> timeout reports.
> >> If you wish, I can send you console logs directly.
> >>
> >> However as far as I understand i2o layer does not have any
> >> sort of timeout/error
> >> handlers (I hope Markus correct me if I'm err), and it would
> >> be great if this
> >> feature will be appear in the future.
> >>
> >>> If it will help, Vasily, contact me for the latest
> dpt_i2o driver as
> >>> that is the driver I am most familiar with; it may be of
> interest to
> >>> determine if the problem duplicates with the dpt_i2o
> driver. Keep in
> >>> mind that the i2o driver is a block driver, dpt_i2o is a
> >> scsi driver.

> >>
> >> Unfortunately we do not know how we can reproduce this issue.
> >> Currently it
> >> occurs on the production nodes only and customers are very
> >> against of any
> >> experiments on these nodes.
> >>
> >> Therefore it is not to easy to switch from i2o layer to your
> >> dpt_i2o driver.
> >>
> >> Currently we have not dpt_i2o driver in our kernels. The most
> >> important reasons are:
> >> - this driver did have some problems on 64-bit kernels (but
> >> it is resolved
> >> already, I'm I right?).
> >> - it is not included into 2.6-based Red Hat distributiuons.
> >> - it did not worked when I've tried to compile it into kernel.
> >> - when I've tried to build it as module, I've discovered that
> >> it conflicts with
> >> i2o drivers: initscripts on the some distributions (FC4?)
> >> have tried to load
> >> both of these modules (one from initrd, second -- when
> >> detects according PCIID)
> >> and it hangs the node. I've not found any working combination
> >> and therefore
> >> we've decided to not include dpt_i2o driver into our 2.6 kernels.
> >>
> >> However, Mark, I'm ready to check your new driver on our
> >> internal testnodes, and
> >> if last issue (modules conflicts) is fixed I'll try to
> >> include your driver into
> >> our kernels.
> >>
> >> Thank you,
> >> Vasily Averin
> >>
> >>> Sincerely -- Mark Salyzyn
> >>>
> >>>> -----Original Message-----
> >>>> From: linux-scsi-owner@vger.kernel.org
> >>>> [mailto:linux-scsi-owner@vger.kernel.org] On Behalf Of
> Vasily Averin
> >>>> Sent: Friday, August 04, 2006 7:50 AM
> >>>> To: linux-scsi@vger.kernel.org; Markus Lidel
> >>>> Cc: devel@openvz.org
> >>>> Subject: i2o hardware hangs (ASR-2010S)
> >>>>
> >>>>

> >>>> Hello Markus,
> >>>>
> >>>> We experience problems with I2O hardware on 2.6 kernels,
> >>>> probably this can help
> >>>> you or maybe you even know the answer. Can you please,
> take a look?
> >>>>
> >>>> After migration to 2.6 kernels our customers began to claim
> >>>> that i2o-based
> >>>> nodes hang. We have investigated these claims and discovered
> >>>> that i2o disks on
> >>>> theses nodes stopped the processing of any IO requests.
> >>>> Please, note, it is not
> >>>> a single issue, it happens from time to time.
> >>>>
> >>>> Our kernel-space watchdog module has produced the following
> >>>> output to serial console
> >>>>
> >>>> Jul 31 07:38:37
> >>>> (80,0) i2o/hda r(77135616 1632632476 15538880) w(69903626
> >>>> 1034743472 407332291)
> >>>> Jul 31 07:39:38
> >>>> (80,0) i2o/hda r(77148190 1633252850 15543968) w(69906364
> >>>> 1034764548 407338084)
> >>>> (80,0) i2o/hda r(77157038 1633672916 15546672) w(69912375
> >>>> 1034808048 407351490)
> >>>> (80,0) i2o/hda r(77169933 1634285356 15550897) w(69916317
> >>>> 1034845588 407364374)
> >>>> (80,0) i2o/hda r(77178290 1634941276 15555039) w(69919031
> >>>> 1034865212 407369386)
> >>>> (80,0) i2o/hda r(77192170 1635427776 15559925) w(69922676
> >>>> 1034892406 407377617)
> >>>> (80,0) i2o/hda r(77216478 1635774384 15570783) w(69927294
> >>>> 1034921708 407385382)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928376 407387163)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928378 407387163)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928384 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928384 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928384 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928386 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928390 407387164)

> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928390 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928390 407387164)
> >>>> (80,0) i2o/hda r(77221642 1635925752 15572389) w(69927966
> >>>> 1034928390 407387164)
> >>>>
> >>>> where r(reads, read_sectors, read_merges) w(writes,
> >>>> write_sectors, write_merges)
> >>>>
> >>>> Magic keys works, according to showProcess processors are in
> >>>> idle, ShowTraces
> >>>> shows a few thousand processes in D-state, but we can not
> >>>> find any deadlocks, it
> >>>> looks like the processes waits until I/O finished.
> >>>> Unfortunately i2o layer has
> >>>> no any error handlers and there is no any chance that the
> >>>> node will return
> >>> >from this coma.
> >>>> Described incident has occurred after ~2 weeks uptime. It was
> >>>> Supermicro X5DP8
> >>>> motherboard /8Gb memory /Adaptec ASR-2010S I2O Zero
> Channel. Kernel
> >>>> 2.6.8-022stab078.9-enterprise, sources/configs are accessible
> >>>> on openvz.org.
> >>>>
> >>>> In the bootlogs I've found mtrr message. As far as I know you
> >>>> have fixed this
> >>>> issue, however I'm not sure that it can leads to described hang.
> >>>>
> >>>> I2O Core - (C) Copyright 1999 Red Hat Software
> >>>> i2o: max_drivers=4
> >>>> i2o: Checking for PCI I2O controllers...
> >>>> ACPI: PCI interrupt 0000:06:01.0[A] -> GSI 72 (level,
> low) -> IRQ 72
> >>>> i2o: I2O controller found on bus 6 at 8.
> >>>> i2o: PCI I2O controller
> >>>>   BAR0 at 0xF8400000 size=1048576
> >>>>   BAR1 at 0xFB000000 size=16777216
> >>>> mtrr: type mismatch for fb000000,1000000 old: uncachable new:
> >>>> write-combining
> >>>> i2o: could not enable write combining MTRR
> >>>> iop0: Installed at IRQ 72
> >>>> iop0: Activating I2O controller...
> >>>> iop0: This may take a few minutes if there are many devices
> >>>> iop0: HRT has 1 entries of 16 bytes each.
> >>>> Adapter 00000012: TID 0000:[HPC*]:PCI 1: Bus 1 Device 22
> Function 0

> >>>> iop0: Controller added
> >>>> I2O Block Storage OSM v0.9
> >>>>   (c) Copyright 1999-2001 Red Hat Software.
> >>>> block-osm: registered device at major 80
> >>>> block-osm: New device detected (TID: 211)
> >>>> Using anticipatory io scheduler
> >>>> i2o/hda: i2o/hda1 i2o/hda2 < i2o/hda5 i2o/hda6 >
> >>>>
> >>>> # cat /proc/mtrr
> >>>> reg00: base=0xf8000000 (3968MB), size= 128MB: uncachable, count=1
> >>>> reg01: base=0x00000000 (0MB), size=8192MB: write-back, count=1
> >>>> reg02: base=0x200000000 (8192MB), size= 128MB:
> write-back, count=1
> >>>> reg03: base=0xf7f80000 (3967MB), size= 512KB: uncachable, count=1
> >>>>
> >>>> I would repeat, it is not a single fault, we have received
> >>>> similar claims once
> >>>> and again. For some time we believed that it was due some
> >>>> hardware faults,
> >>>> however some doubts are cast upon it. The same nodes worked
> >>>> well long time ago
> >>>> without any troubles under 2.4-based kernels with dpt_i2o
> >>>> driver and we have not
> >>>> observed any of i2o hardware troubles so frequently.
> >>>>
> >>>> Is it possible that our kernel (based on 2.6.8.1 mainstream)
> >>>> have some bugs in
> >>>> i2o drivers? However we're using driver sources taken from
> >>>> RHEL4U2 kernel, and I
> >>>> cannot find any similar claims from RHEL4 customers.
> >>>>
> >>>> Is it possible than we have some other related kernels bugs?
> >>>> In this case why we
> >>>> have such kind of issues only on i2o-based nodes?
> >>>>
> >>>> Could you please give me some hints which allow me to
> >>>> continue investigation of
> >>>> this issue. If you have any suggestions I'll check them
> next time.
> >>>>
> >>>> Thank you,
> >>>>   Vasily Averin
> >>>>
> >>>> SWsoft Virtuozzo/OpenVZ Linux kernel team
>
>
> Best regards,
>

>
> Markus Lidel
> ----------
> Markus Lidel (Senior IT Consultant)
>
> Shadow Connect GmbH
> Carl-Reisch-Weg 12
> D-86381 Krumbach
> Germany
>
> Phone: +49 82 82/99 51-0
> Fax: +49 82 82/99 51-11
>
> E-Mail: Markus.Lidel@shadowconnect.com
> URL: http://www.shadowconnect.com
>

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Tue, 08 Aug 2006 08:12:04 GMT
View Forum Message <> Reply to Message

Markus Lidel wrote:
>> However, Mark, I'm ready to check your new driver on our internal
>> testnodes, and
>> if last issue (modules conflicts) is fixed I'll try to include your
>> driver into
>> our kernels.
>
> With the latest version the module conflicts should be fixed.

Markus, Mark,

thank you, I've found your patches in
http://bugzilla.kernel.org/show_bug.cgi?id=4940

Thank you,
 Vasily Averin

SWsoft Virtuozzo/OpenVZ Linux kernel team

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Tue, 08 Aug 2006 09:47:57 GMT
View Forum Message <> Reply to Message

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

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by mark_salyzyn on Tue, 08 Aug 2006 12:44:42 GMT
View Forum Message <> Reply to Message

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
>

File Attachments

1) dpt_i2o-2.5.0-2426.tgz, downloaded 520 times

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by Markus Lidel on Tue, 08 Aug 2006 21:55:54 GMT
View Forum Message <> Reply to Message

Hello,

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.

Hmmm, personally i would only add error recovery if the behaviour
couldn't be solved otherway (in this case the problem is solved already
in recent kernels), because the controller should already handle it
(regarding to the I2O spec). But if it is wanted i would add it.

> 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).
>> -----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)
>> 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.

Hmmm, the 2.6.8 kernel is very old in terms of my work. The changes made to this kernel where just to get something working at all. In more recent kernels (expect 2.6.16, which is broken) it should work fine without the hangup (in the early versions of the kernel the messages transfered to the controller was to large, which lead to the hangup you reported). I would suggest at least 2.6.13 if possible.

>>> 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.

Best regards,

Markus Lidel

Markus Lidel (Senior IT Consultant)

Shadow Connect GmbH
Carl-Reisch-Weg 12
D-86381 Krumbach
Germany

Phone: +49 82 82/99 51-0
Fax: +49 82 82/99 51-11

E-Mail: Markus.Lidel@shadowconnect.com
URL: http://www.shadowconnect.com

Subject: Re: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Mon, 14 Aug 2006 14:02:19 GMT
View Forum Message <> Reply to Message

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
>

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by mark_salyzyn on Mon, 14 Aug 2006 14:28:47 GMT
View Forum Message <> Reply to Message

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);
         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
> >
>
>

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
```

Subject: RE: i2o hardware hangs (ASR-2010S)
Posted by Dale Bohl on Mon, 11 Dec 2006 22:08:43 GMT
View Forum Message <> Reply to Message

TWIMC,

Is it possible for someone to send me the patch for the Adaptec Zero channel Raid controller 2010s? I've tried nearly everything and I'd really like to use HW raid versus SW raid on this superMicro piece of crap.

Dale Bohl - Senior Systems Administrator
Information Systems
Mason Companies, Inc.
dbohl@masonshoe.com
715-720-4382

Subject: Re: RE: i2o hardware hangs (ASR-2010S)
Posted by vaverin on Tue, 12 Dec 2006 08:32:10 GMT
View Forum Message <> Reply to Message

Dale,

I would say that the troubles with i2o hardware went away after i2o layer
update. I've backported i2o sources from 2.6.17 mainline kernel and it seems it
works well, we do not have any new customers complains now.

You can find the patch in our 2.6.8-022stab078.21 kernel sources:

http://openvz.org/news/updates/kernel-022stab078.21

* linux-2.6.8-i2o-1.325.patch:
  Patch from Vasily (vvs@):
  updates i2o layer, backported from to 2.6.17 linux mainstream kernel

Thank you,
    Vasily Averin

Dale Bohl wrote:
> TWIMC,
>
>   Is it possible for someone to send me the patch for the Adaptec Zero channel
> Raid controller 2010s? I've tried nearly everything and I'd really like to
> use HW raid versus SW raid on this superMicro piece of crap.
>
>
>
> --
> Dale Bohl - Senior Systems Administrator
> Information Systems
> Mason Companies, Inc.
> dbohl@masonshoe.com
> 715-720-4382