Subject: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by den on Fri, 03 Aug 2007 10:39:24 GMT
View Forum Message <> Reply to Message

The following calltrace is possible now:
```
 handle_sysrq
   machine_emergency_restart
     mach_reboot_fixups
       pci_get_device
         pci_get_subsys
    down_read
```
The patch obtains PCI device during initialization to avoid bothering PCI
search engine in interrupt. Devices used in this code are not supposed to
be pluggable, so it looks safe to keep them.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
diff --git a/arch/i386/kernel/reboot_fixups.c b/arch/i386/kernel/reboot_fixups.c
index 03e1cce..873ad55 100644
--- a/arch/i386/kernel/reboot_fixups.c
+++ b/arch/i386/kernel/reboot_fixups.c
@@ -37,6 +37,7 @@ struct device_fixup {
  unsigned int vendor;
  unsigned int device;
  void (*reboot_fixup)(struct pci_dev *);
+ struct pci_dev *dev;
 };

 static struct device_fixup fixups_table[] = {
@@ -49,20 +50,35 @@ static struct device_fixup fixups_table[] = {
  * is a fixup, we call it and we expect to never return from it. if we
  * do return, we keep looking and then eventually fall back to the
  * standard mach_reboot on return.
+ *
+ * Unfortunately, this code can be called from an interrupt and it is
+ * impossible to get PCI device directly. So, lets prepare the list
+ * beforehand.
  */
 void mach_reboot_fixups(void)
 {
  struct device_fixup *cur;
- struct pci_dev *dev;
  int i;

  for (i=0; i < ARRAY_SIZE(fixups_table); i++) {
   cur = &(fixups_table[i]);
-  dev = pci_get_device(cur->vendor, cur->device, NULL);
```

```
-  if (!dev)
+   if (cur->dev == NULL)
     continue;

-  cur->reboot_fixup(dev);
+   cur->reboot_fixup(cur->dev);
+ }
+}
+
+int mach_fixup_init(void)
+{
+ struct device_fixup *cur;
+ int i;
+
+ for (i=0; i < ARRAY_SIZE(fixups_table); i++) {
+   cur = &(fixups_table[i]);
+   cur->dev = pci_get_device(cur->vendor, cur->device, NULL);
  }
+ return 0;
 }

+module_init(mach_fixup_init);
```

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Greg KH on Sat, 04 Aug 2007 04:08:06 GMT
View Forum Message <> Reply to Message

On Fri, Aug 03, 2007 at 02:39:24PM +0400, Denis V. Lunev wrote:
> The following calltrace is possible now:
> handle_sysrq
>   machine_emergency_restart
>   mach_reboot_fixups
>   pci_get_device
>   pci_get_subsys
>   down_read
> The patch obtains PCI device during initialization to avoid bothering PCI
> search engine in interrupt. Devices used in this code are not supposed to
> be pluggable, so it looks safe to keep them.

What devices are supposed to be affected here? Are you sure that they
can't be removed later? Grabbing references here might mess with them
in the future.

thanks,

greg k-h

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by den on Mon, 06 Aug 2007 07:16:20 GMT
View Forum Message <> Reply to Message

Greg KH wrote:
> On Fri, Aug 03, 2007 at 02:39:24PM +0400, Denis V. Lunev wrote:
>> The following calltrace is possible now:
>> handle_sysrq
>>   machine_emergency_restart
>>   mach_reboot_fixups
>>   pci_get_device
>>   pci_get_subsys
>>   down_read
>> The patch obtains PCI device during initialization to avoid bothering PCI
>> search engine in interrupt. Devices used in this code are not supposed to
>> be pluggable, so it looks safe to keep them.
>
> What devices are supposed to be affected here? Are you sure that they
> can't be removed later? Grabbing references here might mess with them
> in the future.
Right now the list is the following:
static struct device_fixup fixups_table[] = {
{ PCI_VENDOR_ID_CYRIX, PCI_DEVICE_ID_CYRIX_5530_LEGACY,
cs5530a_warm_reset },
{ PCI_VENDOR_ID_AMD, PCI_DEVICE_ID_AMD_CS5536_ISA, cs5536_warm_reset },
};

Though, if the approach is not suitable, we can skip fixups if we came
from sysrq.

Regards,
 Den

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Andrew Morton on Mon, 06 Aug 2007 20:03:26 GMT
View Forum Message <> Reply to Message

On Fri, 3 Aug 2007 14:39:24 +0400 "Denis V. Lunev" <den@openvz.org> wrote:

> The following calltrace is possible now:
> handle_sysrq
>   machine_emergency_restart
>   mach_reboot_fixups
>   pci_get_device
>   pci_get_subsys
>   down_read
> The patch obtains PCI device during initialization to avoid bothering PCI

> search engine in interrupt. Devices used in this code are not supposed to
> be pluggable, so it looks safe to keep them.
>

hm.

>
> diff --git a/arch/i386/kernel/reboot_fixups.c b/arch/i386/kernel/reboot_fixups.c
> index 03e1cce..873ad55 100644
> --- a/arch/i386/kernel/reboot_fixups.c
> +++ b/arch/i386/kernel/reboot_fixups.c
> @@ -37,6 +37,7 @@ struct device_fixup {
> unsigned int vendor;
> unsigned int device;
> void (*reboot_fixup)(struct pci_dev *);
> + struct pci_dev *dev;
> };
>
> static struct device_fixup fixups_table[] = {
> @@ -49,20 +50,35 @@ static struct device_fixup fixups_table[] = {
> * is a fixup, we call it and we expect to never return from it. if we
> * do return, we keep looking and then eventually fall back to the
> * standard mach_reboot on return.
> + *
> + * Unfortunately, this code can be called from an interrupt and it is
> + * impossible to get PCI device directly. So, lets prepare the list
> + * beforehand.

This comment should tell the reader which interrupt path that is (ie: sysrq-B).

> */
> void mach_reboot_fixups(void)
> {
> struct device_fixup *cur;
> - struct pci_dev *dev;
> int i;
>
> for (i=0; i < ARRAY_SIZE(fixups_table); i++) {
> cur = &(fixups_table[i]);
> - dev = pci_get_device(cur->vendor, cur->device, NULL);
> - if (!dev)
> + if (cur->dev == NULL)
> continue;
>
> - cur->reboot_fixup(dev);
> + cur->reboot_fixup(cur->dev);
> + }
> +}

```
> +
> +int mach_fixup_init(void)
> +{
> + struct device_fixup *cur;
> + int i;
> +
> + for (i=0; i < ARRAY_SIZE(fixups_table); i++) {
> +  cur = &(fixups_table[i]);
> +  cur->dev = pci_get_device(cur->vendor, cur->device, NULL);
>   }
> + return 0;
>  }
>
> +module_init(mach_fixup_init);
```

I'm not sure that we want to make core PCI code capable of being called
from interrupt context just for the sake of sysrq-B. It adds complexity
and maintenance hassles for something which is largely a debugging feature.

otoh, the patch is faily simple-looking and people _do_ use sysrq-B fairly
often so I guess we'll find out if we break it again.

otoh2, perhaps we can find some quicky hack on the sysrq patch to shut up
the might_sleep() warnings (which I presume is the only problem which is
presently being exhibited?). Something like the unpleasant
oops_in_progress, perhaps.

Greg, any preferences?

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Greg KH on Tue, 07 Aug 2007 02:49:10 GMT
View Forum Message <> Reply to Message

On Mon, Aug 06, 2007 at 11:16:20AM +0400, Denis V. Lunev wrote:
> Greg KH wrote:
> > On Fri, Aug 03, 2007 at 02:39:24PM +0400, Denis V. Lunev wrote:
> >> The following calltrace is possible now:
> >> handle_sysrq
> >>   machine_emergency_restart
> >>   mach_reboot_fixups
> >>   pci_get_device
> >>   pci_get_subsys
> >>   down_read
> >> The patch obtains PCI device during initialization to avoid bothering PCI
> >> search engine in interrupt. Devices used in this code are not supposed to
> >> be pluggable, so it looks safe to keep them.
> >

> > What devices are supposed to be affected here? Are you sure that they
> > can't be removed later? Grabbing references here might mess with them
> > in the future.
> Right now the list is the following:
> static struct device_fixup fixups_table[] = {
> { PCI_VENDOR_ID_CYRIX, PCI_DEVICE_ID_CYRIX_5530_LEGACY,
> cs5530a_warm_reset },
> { PCI_VENDOR_ID_AMD, PCI_DEVICE_ID_AMD_CS5536_ISA, cs5536_warm_reset },
> };
>
> Though, if the approach is not suitable, we can skip fixups if we came
> from sysrq.

I don't think we really need to do fixups when we are "crashing" like this. The user really isn't shutting down the kernel as it should normally do.

Andrew, I really don't want to change the PCI core to handle this, as we finally fixed a lot of issues with drivers trying to walk these lists from interrupt context. So if you want to just hide the warning message as we are shutting down, that's fine with me. Or just don't do the fixups. But grabbing a reference to the pci device is unsafe in my opinion and I do not want to do that.

thanks,

greg k-h

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Andrew Morton on Tue, 07 Aug 2007 07:24:37 GMT
View Forum Message <> Reply to Message

On Mon, 6 Aug 2007 19:49:10 -0700 Greg KH <gregkh@suse.de> wrote:

> On Mon, Aug 06, 2007 at 11:16:20AM +0400, Denis V. Lunev wrote:
> > Greg KH wrote:
> > > On Fri, Aug 03, 2007 at 02:39:24PM +0400, Denis V. Lunev wrote:
> > >> The following calltrace is possible now:
> > >> handle_sysrq
> > >> machine_emergency_restart
> > >> mach_reboot_fixups
> > >> pci_get_device
> > >> pci_get_subsys
> > >> down_read
> > >> The patch obtains PCI device during initialization to avoid bothering PCI
> > >> search engine in interrupt. Devices used in this code are not supposed to
> > >> be pluggable, so it looks safe to keep them.

> > >
> > > What devices are supposed to be affected here? Are you sure that they
> > > can't be removed later? Grabbing references here might mess with them
> > > in the future.
> > Right now the list is the following:
> > static struct device_fixup fixups_table[] = {
> > { PCI_VENDOR_ID_CYRIX, PCI_DEVICE_ID_CYRIX_5530_LEGACY,
> > cs5530a_warm_reset },
> > { PCI_VENDOR_ID_AMD, PCI_DEVICE_ID_AMD_CS5536_ISA, cs5536_warm_reset },
> > };
> >
> > Though, if the approach is not suitable, we can skip fixups if we came
> > from sysrq.
>
> I don't think we really need to do fixups when we are "crashing" like
> this. The user really isn't shutting down the kernel as it should
> normally do.
>
> Andrew, I really don't want to change the PCI core to handle this, as we
> finally fixed a lot of issues with drivers trying to walk these lists
> from interrupt context. So if you want to just hide the warning message
> as we are shutting down, that's fine with me. Or just don't do the
> fixups. But grabbing a reference to the pci device is unsafe in my
> opinion and I do not want to do that.
>

OK, good decision ;)

One approach would be for some brave soul to pick his way through
the reboot code and ensure that we are correctly and reliably setting
system_state to SYSTEM_RESTART, then test that in __might_sleep().

But this does suppress somewhat-useful debugging just because of sysrq-B
and I really wouldn't want to utilise the horrid system_state any more that
we are presently doing. I think on balance that it would be better if we
could do something more targetted, like modify emergency_restart() to test
in_interrupt() and to then apologetically set some well-named global flag
which will shut up __might_sleep(). Pretty foul, but I can't think of
anything better.

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Greg KH on Tue, 07 Aug 2007 07:42:32 GMT
View Forum Message <> Reply to Message

On Tue, Aug 07, 2007 at 12:44:55AM -0700, Andrew Morton wrote:
> On Tue, 7 Aug 2007 00:24:37 -0700 Andrew Morton <akpm@linux-foundation.org> wrote:
>

> > > Andrew, I really don't want to change the PCI core to handle this, as we
> > > finally fixed a lot of issues with drivers trying to walk these lists
> > > from interrupt context. So if you want to just hide the warning message
> > > as we are shutting down, that's fine with me. Or just don't do the
> > > fixups. But grabbing a reference to the pci device is unsafe in my
> > > opinion and I do not want to do that.
> > >
> >
> > OK, good decision ;)
> >
> > One approach would be for some brave soul to pick his way through
> > the reboot code and ensure that we are correctly and reliably setting
> > system_state to SYSTEM_RESTART, then test that in __might_sleep().
> >
> > But this does suppress somewhat-useful debugging just because of sysrq-B
> > and I really wouldn't want to utilise the horrid system_state any more that
> > we are presently doing. I think on balance that it would be better if we
> > could do something more targetted, like modify emergency_restart() to test
> > in_interrupt() and to then apologetically set some well-named global flag
> > which will shut up __might_sleep(). Pretty foul, but I can't think of
> > anything better.
>
> ok, this might be better. How about we just stop calling mach_reboot_fixups()
> at sysrq-B time?

Fine with me, but what hardware will be messed up because of this?

thanks,

greg k-h

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by Andrew Morton on Tue, 07 Aug 2007 07:44:55 GMT
View Forum Message <> Reply to Message

On Tue, 7 Aug 2007 00:24:37 -0700 Andrew Morton <akpm@linux-foundation.org> wrote:

> > Andrew, I really don't want to change the PCI core to handle this, as we
> > finally fixed a lot of issues with drivers trying to walk these lists
> > from interrupt context. So if you want to just hide the warning message
> > as we are shutting down, that's fine with me. Or just don't do the
> > fixups. But grabbing a reference to the pci device is unsafe in my
> > opinion and I do not want to do that.
> >
>
> OK, good decision ;)
>

> One approach would be for some brave soul to pick his way through
> the reboot code and ensure that we are correctly and reliably setting
> system_state to SYSTEM_RESTART, then test that in __might_sleep().
>
> But this does suppress somewhat-useful debugging just because of sysrq-B
> and I really wouldn't want to utilise the horrid system_state any more that
> we are presently doing. I think on balance that it would be better if we
> could do something more targetted, like modify emergency_restart() to test
> in_interrupt() and to then apologetically set some well-named global flag
> which will shut up __might_sleep(). Pretty foul, but I can't think of
> anything better.

ok, this might be better. How about we just stop calling mach_reboot_fixups()
at sysrq-B time?

> > >> handle_sysrq
> > >>   machine_emergency_restart
> > >>   mach_reboot_fixups
> > >>   pci_get_device
> > >>   pci_get_subsys
> > >>   down_read

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by den on Tue, 07 Aug 2007 07:48:29 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Mon, 6 Aug 2007 19:49:10 -0700 Greg KH <gregkh@suse.de> wrote:
>
>> On Mon, Aug 06, 2007 at 11:16:20AM +0400, Denis V. Lunev wrote:
>>> Greg KH wrote:
>>>> On Fri, Aug 03, 2007 at 02:39:24PM +0400, Denis V. Lunev wrote:
>>>>> The following calltrace is possible now:
>>>>> handle_sysrq
>>>>>   machine_emergency_restart
>>>>>   mach_reboot_fixups
>>>>>   pci_get_device
>>>>>   pci_get_subsys
>>>>>   down_read
>>>>> The patch obtains PCI device during initialization to avoid bothering PCI
>>>>> search engine in interrupt. Devices used in this code are not supposed to
>>>>> be pluggable, so it looks safe to keep them.
>>>> What devices are supposed to be affected here? Are you sure that they
>>>> can't be removed later? Grabbing references here might mess with them
>>>> in the future.
>>> Right now the list is the following:
>>> static struct device_fixup fixups_table[] = {

>>> { PCI_VENDOR_ID_CYRIX, PCI_DEVICE_ID_CYRIX_5530_LEGACY,
>>> cs5530a_warm_reset },
>>> { PCI_VENDOR_ID_AMD, PCI_DEVICE_ID_AMD_CS5536_ISA, cs5536_warm_reset },
>>> };
>>>
>>> Though, if the approach is not suitable, we can skip fixups if we came
>>> from sysrq.
>> I don't think we really need to do fixups when we are "crashing" like
>> this. The user really isn't shutting down the kernel as it should
>> normally do.
>>
>> Andrew, I really don't want to change the PCI core to handle this, as we
>> finally fixed a lot of issues with drivers trying to walk these lists
>> from interrupt context. So if you want to just hide the warning message
>> as we are shutting down, that's fine with me. Or just don't do the
>> fixups. But grabbing a reference to the pci device is unsafe in my
>> opinion and I do not want to do that.
>>
>
> OK, good decision ;)
>
> One approach would be for some brave soul to pick his way through
> the reboot code and ensure that we are correctly and reliably setting
> system_state to SYSTEM_RESTART, then test that in __might_sleep().
>
> But this does suppress somewhat-useful debugging just because of sysrq-B
> and I really wouldn't want to utilise the horrid system_state any more that
> we are presently doing. I think on balance that it would be better if we
> could do something more targetted, like modify emergency_restart() to test
> in_interrupt() and to then apologetically set some well-named global flag
> which will shut up __might_sleep(). Pretty foul, but I can't think of
> anything better.

__might_sleep prevention will solve the problem only partially :(There
is a direct WARN_ON(in_interrupt()) in pci_get_subsys.

IMHO, calling down_read(&pci_bus_sem); from sysrq-B is not an option.
I'll send a fixup disabling patch in a moment.

Subject: Re: [PATCH] pci_get_device call from interrupt in reboot fixups
Posted by den on Tue, 07 Aug 2007 07:49:52 GMT
View Forum Message <> Reply to Message

Greg KH wrote:
> On Tue, Aug 07, 2007 at 12:44:55AM -0700, Andrew Morton wrote:
>> On Tue, 7 Aug 2007 00:24:37 -0700 Andrew Morton <akpm@linux-foundation.org> wrote:
>>

>>>> Andrew, I really don't want to change the PCI core to handle this, as we
>>>> finally fixed a lot of issues with drivers trying to walk these lists
>>>> from interrupt context. So if you want to just hide the warning message
>>>> as we are shutting down, that's fine with me. Or just don't do the
>>>> fixups. But grabbing a reference to the pci device is unsafe in my
>>>> opinion and I do not want to do that.
>>>>
>>> OK, good decision ;)
>>>
>>> One approach would be for some brave soul to pick his way through
>>> the reboot code and ensure that we are correctly and reliably setting
>>> system_state to SYSTEM_RESTART, then test that in __might_sleep().
>>>
>>> But this does suppress somewhat-useful debugging just because of sysrq-B
>>> and I really wouldn't want to utilise the horrid system_state any more that
>>> we are presently doing. I think on balance that it would be better if we
>>> could do something more targetted, like modify emergency_restart() to test
>>> in_interrupt() and to then apologetically set some well-named global flag
>>> which will shut up __might_sleep(). Pretty foul, but I can't think of
>>> anything better.
>> ok, this might be better. How about we just stop calling mach_reboot_fixups()
>> at sysrq-B time?
>
> Fine with me, but what hardware will be messed up because of this?

static struct device_fixup fixups_table[] = {
> > { PCI_VENDOR_ID_CYRIX, PCI_DEVICE_ID_CYRIX_5530_LEGACY,
> > cs5530a_warm_reset },
> > { PCI_VENDOR_ID_AMD, PCI_DEVICE_ID_AMD_CS5536_ISA, cs5536_warm_reset },
> > };