Subject: [PATCH] Add ability to print calltraces tighter on i386
Posted by Pavel Emelianov on Wed, 08 Aug 2007 14:17:51 GMT
View Forum Message <> Reply to Message

When printing a BUG or OOPS report the longest part of it is
the calltrace, which sometimes (quite often) doesn't fit the
standard 25-lines display. This may become a bad news when the
system doesn't have a serial/net console and is completely
frozen so that the terminal scrolling doesn't work.

The information that hides from the developer is registers, the
top of the calltrace and information about the kernel and the
crashed process (uname). As our experience shows, seeing this
info is sometimes critical and having a short calltrace would
help a lot.

The proposal is to make a boot-option called "tight_trace", that
makes the calltrace show only the addresses in one line instead
of the symbol names one per line.

E.g. OOPSes of 50 lines occupy ~20 with this patch.

This is an example of how it will look for i386, but if this
will be found useful, I will make the patch for other arched
I can test it on (at least x86_64, ia64).

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

 arch/i386/kernel/traps.c | 29 ++++++++++++++++++++++++++----
 1 files changed, 25 insertions(+), 4 deletions(-)

```
diff --git a/arch/i386/kernel/traps.c b/arch/i386/kernel/traps.c
index f469d20..8aa2b3b 100644
--- a/arch/i386/kernel/traps.c
+++ b/arch/i386/kernel/traps.c
@@ -186,6 +186,19 @@ void dump_trace(struct task_struct *task
 }
 EXPORT_SYMBOL(dump_trace);

+static int decode_oops = 1;
+
+static int __init setup_decode_oops(char *str)
+{
+ if (*str)
+  return 0;
+
```

```
+ decode_oops = 0;
+ return 1;
+}
+
+__setup("tight_trace", setup_decode_oops);
+
 static void
 print_trace_warning_symbol(void *data, char *msg, unsigned long symbol)
 {
@@ -209,8 +222,11 @@ static int print_trace_stack(void *data,
  */
 static void print_trace_address(void *data, unsigned long addr)
 {
- printk("%s [<%08lx>] ", (char *)data, addr);
- print_symbol("%s\n", addr);
+ if (decode_oops) {
+  printk("%s [<%08lx>] ", (char *)data, addr);
+  print_symbol("%s\n", addr);
+ } else
+  printk("[<%08lx>] ", addr);
  touch_nmi_watchdog();
 }

@@ -226,7 +242,9 @@ show_trace_log_lvl(struct task_struct *t
      unsigned long * stack, char *log_lvl)
 {
  dump_trace(task, regs, stack, &print_trace_ops, log_lvl);
- printk("%s =======================\n", log_lvl);
+ if (decode_oops)
+  printk("%s =======================", log_lvl);
+ printk("\n");
 }

 void show_trace(struct task_struct *task, struct pt_regs *regs,
@@ -256,7 +274,10 @@ static void show_stack_log_lvl(struct ta
    printk("\n%s       ", log_lvl);
   printk("%08lx ", *stack++);
  }
- printk("\n%sCall Trace:\n", log_lvl);
+ printk("\n%sCall Trace: ", log_lvl);
+ if (decode_oops)
+  printk("\n");
+
  show_trace_log_lvl(task, regs, esp, log_lvl);
  debug_show_held_locks(task);
 }
```

Subject: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by Andi Kleen on Wed, 08 Aug 2007 14:32:09 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov <xemul@openvz.org> writes:

> When printing a BUG or OOPS report the longest part of it is
> the calltrace, which sometimes (quite often) doesn't fit the
> standard 25-lines display. This may become a bad news when the
> system doesn't have a serial/net console and is completely frozen so
> that the terminal scrolling doesn't work.
>
> The information that hides from the developer is registers, the
> top of the calltrace and information about the kernel and the
> crashed process (uname). As our experience shows, seeing this info is
> sometimes critical and having a short calltrace would help a lot.
>
> The proposal is to make a boot-option called "tight_trace", that
> makes the calltrace show only the addresses in one line instead
> of the symbol names one per line.
>
> E.g. OOPSes of 50 lines occupy ~20 with this patch.
>
> This is an example of how it will look for i386, but if this
> will be found useful, I will make the patch for other arched
> I can test it on (at least x86_64, ia64).

Just use a higher resolution with vga=...
I have yet to see an oops that doesn't fit on 80x50

-Andi

Subject: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by Pavel Emelianov on Wed, 08 Aug 2007 15:09:34 GMT
View Forum Message <> Reply to Message

Andi Kleen wrote:
> Pavel Emelyanov <xemul@openvz.org> writes:
>
>> When printing a BUG or OOPS report the longest part of it is
>> the calltrace, which sometimes (quite often) doesn't fit the
>> standard 25-lines display. This may become a bad news when the
>> system doesn't have a serial/net console and is completely frozen so
>> that the terminal scrolling doesn't work.
>>
>> The information that hides from the developer is registers, the
>> top of the calltrace and information about the kernel and the

>> crashed process (uname). As our experience shows, seeing this info is
>> sometimes critical and having a short calltrace would help a lot.
>>
>> The proposal is to make a boot-option called "tight_trace", that
>> makes the calltrace show only the addresses in one line instead
>> of the symbol names one per line.
>>
>> E.g. OOPSes of 50 lines occupy ~20 with this patch.
>>
>> This is an example of how it will look for i386, but if this
>> will be found useful, I will make the patch for other arched
>> I can test it on (at least x86_64, ia64).
>
> Just use a higher resolution with vga=...
> I have yet to see an oops that doesn't fit on 80x50

Not everyone likes frame buffer but even with it any OOPs in
network code which happens in softirq, io scheduler and nearby
code that is called after passing through all the VFS hooks
and many other examples produce long oopses.

Oops-es with only the calltrace of ~50 lines do happen :)

> -Andi
> -
> To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
> Please read the FAQ at http://www.tux.org/lkml/
>

Subject: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by Andi Kleen on Wed, 08 Aug 2007 15:20:08 GMT
View Forum Message <> Reply to Message

> Not everyone likes frame buffer

You don't need the frame buffer; cards typically have text mode
fonts upto 80x50. The node numbers vary, but you can find out yours
with vga=ask

> but even with it any OOPs in
> network code which happens in softirq, io scheduler and nearby
> code that is called after passing through all the VFS hooks
> and many other examples produce long oopses.
>
> Oops-es with only the calltrace of ~50 lines do happen :)

Normally most of it bogus. I had hoped to address this with the dwarf2
unwinder, which tends to filter them out nicely,
but Linus unfortunately has developed an quite irrational aversion against it and
it's not in.

But the problem is with bogus entries in there you have no guarantee
that the first of your call trace is any useful -- it might be all bogus.
So i don't really think your option makes much sense.

Another way would be to not dump addresses and use multiple entries
per line again. I guess that would make more sense as an option.

-Andi

Subject: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by philipp.marek on Thu, 09 Aug 2007 06:04:23 GMT
View Forum Message <> Reply to Message

On Mittwoch, 8. August 2007, Andi Kleen wrote:
> > Not everyone likes frame buffer
>
> You don't need the frame buffer; cards typically have text mode
> fonts upto 80x50. The node numbers vary, but you can find out yours
> with vga=ask
>
> > but even with it any OOPs in
> > network code which happens in softirq, io scheduler and nearby
> > code that is called after passing through all the VFS hooks
> > and many other examples produce long oopses.
> >
> > Oops-es with only the calltrace of ~50 lines do happen :)
>
> Normally most of it bogus. I had hoped to address this with the dwarf2
> unwinder, which tends to filter them out nicely,
> but Linus unfortunately has developed an quite irrational aversion against
> it and it's not in.
>
> But the problem is with bogus entries in there you have no guarantee
> that the first of your call trace is any useful -- it might be all bogus.
> So i don't really think your option makes much sense.
>
> Another way would be to not dump addresses and use multiple entries
> per line again. I guess that would make more sense as an option.
Maybe a crazy idea, but if some OOPS is reproduceable, it might work to dump
some subset of the calltrace - every time some other part (randomly).

Eg. If you need to show the last 64 stack words (of 8 characters each), but have only space for 80 characters (ie. 8 words in hexadecimal, plus 8 spaces), you could print some index, and every 8th word from there on ...

Or, to increase the probability of getting information for each try, probably use some random steps, too.

Of course, all of this is moot if you've got a serial console ...

Regards,

Phil

Subject: Re: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by dev on Thu, 09 Aug 2007 08:07:26 GMT
View Forum Message <> Reply to Message

Andi Kleen wrote:
>>Not everyone likes frame buffer
>
>
> You don't need the frame buffer; cards typically have text mode
> fonts upto 80x50. The node numbers vary, but you can find out yours
> with vga=ask
>
>
>>but even with it any OOPs in
>>network code which happens in softirq, io scheduler and nearby
>>code that is called after passing through all the VFS hooks
>>and many other examples produce long oopses.
>>
>>Oops-es with only the calltrace of ~50 lines do happen :)
>
>
> Normally most of it bogus. I had hoped to address this with the dwarf2
> unwinder, which tends to filter them out nicely,
> but Linus unfortunately has developed an quite irrational aversion against it and
> it's not in.

Most - but not *all*.
Actually I quite agree with Linus - unwinder is just a pain,
which is the more unreliable then a plain call trace.
Plain call trace has one advantage - it prints more then needed
but it always print the required and clear info.

unwinder goes totally mad when something serious happens like stack
overflows/corruption or other bad thing. 2 my cents.

> But the problem is with bogus entries in there you have no guarantee
> that the first of your call trace is any useful -- it might be all bogus.
> So i don't really think your option makes much sense.

no. bogus entries don't make call trace irrelevant.
And it is very easy to find relevant call trace entries in std output -
call trace should always be correct from the top and from the bottom,
all other entries are checked by eip following the calls.

> Another way would be to not dump addresses and use multiple entries
> per line again. I guess that would make more sense as an option.

Thanks,
Kirill

Subject: Re: [PATCH] Add ability to print calltraces tighter on i386
Posted by Pavel Machek on Tue, 14 Aug 2007 07:11:48 GMT
View Forum Message <> Reply to Message

Hi!

> > E.g. OOPSes of 50 lines occupy ~20 with this patch.
> >
> > This is an example of how it will look for i386, but if this
> > will be found useful, I will make the patch for other arched
> > I can test it on (at least x86_64, ia64).
>
> Just use a higher resolution with vga=...
> I have yet to see an oops that doesn't fit on 80x50

vga= does not work properly in some setups I'd like to debug, like
kexec.
Pavel
--
(english) http://www.livejournal.com/~pavelmachek
(cesky, pictures) http://atrey.karlin.mff.cuni.cz/~pavel/picture/horses/blog.html

---