Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by den on Thu, 22 Mar 2007 10:45:46 GMT
View Forum Message <> Reply to Message

Kirill Korotaev wrote:
> Eric W. Biederman wrote:
>> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
>>
>>
>>> Benjamin Thery and I we were looking at this.
>>> For the moment we are investigating if there is IP fragmentation between the
>>> eth0 and the pair devices.
>>> The profiling shows us "pskb_expand_head" and "csum_partial_copy_generic"
>>> functions have the bigger CPU usage when we are inside a container.
>>
>> Hmm. Interesting hypothesis. I do know I haven't finished the IP fragment
>> support so there may be an issue there. Although generally especially with
>> TCP fragments are not generated. How are you thinking fragmentation would
>> be involved?
>
> I'm not sure I fully understand the test and details of the thread,
> but still...
>
> if network device inside container has MTU higher then eth0 outside the container,
> then packets will get fragmented.
> First time to MTU1 inside container and refragmented to MTU2
> outside the container. At least this is the reason we use ethernet MTU
> on venet devices in OpenVZ - to avoid framentation.

frankly speaking, there are another points, which can call to
pskb_expand_head. Hard header size on container network device should be
the same as on real one.

Regards,
 Den

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Thu, 22 Mar 2007 12:19:09 GMT
View Forum Message <> Reply to Message

"Denis V. Lunev" <den@sw.ru> writes:

> Kirill Korotaev wrote:
>>
>> if network device inside container has MTU higher then eth0 outside the
> container,
>> then packets will get fragmented.

>> First time to MTU1 inside container and refragmented to MTU2
>> outside the container. At least this is the reason we use ethernet MTU
>> on venet devices in OpenVZ - to avoid framentation.
>
> frankly speaking, there are another points, which can call to
> pskb_expand_head. Hard header size on container network device should be
> the same as on real one.

I believe it is the case that I have a 1500 byte mtu with an ethernet header.
Although it may be I didn't reserve extra hard header space or something like
that.

Eric

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by Benjamin Thery on Thu, 22 Mar 2007 17:57:10 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> "Denis V. Lunev" <den@sw.ru> writes:
>
>> Kirill Korotaev wrote:
>>> if network device inside container has MTU higher then eth0 outside the
>> container,
>>> then packets will get fragmented.
>>> First time to MTU1 inside container and refragmented to MTU2
>>> outside the container. At least this is the reason we use ethernet MTU
>>> on venet devices in OpenVZ - to avoid framentation.
>> frankly speaking, there are another points, which can call to
>> pskb_expand_head. Hard header size on container network device should be
>> the same as on real one.
>
> I believe it is the case that I have a 1500 byte mtu with an ethernet header.
> Although it may be I didn't reserve extra hard header space or something like
> that.

My investigations on the increase of cpu load when running netperf
inside a container (ie. through etun2<->etun1) is progressing slowly.

I'm not sure the cause is fragmentation as we supposed initially.
In fact, it seems related to forwarding the packets between the devices.

Here is what I've tracked so far:
* when we run netperf from the container, oprofile reports that the
top "consuming" symbol is: "pskb_expand_head". Next comes
"csum_partial_copy_generic". these symbols represents respectively
13.5% and 9.7% of the samples.

* Without container, these symbols don't show up in the first 20 entries.

Who is calling "pskb_expand_head" in this case?

Using systemtap, I determined that the call to "pskb_expand_head"
comes from the skb_cow() in ip_forward() (l.90 in 2.6.20-rc5-netns).

The number of calls to "pskb_expand_head" matches the number of
invocations of ip_forward() (268000 calls for a 20 seconds netperf
session in my case).

Here is an example of the backtrace when we reach pskb_expand_head:

```
 0xc028bb95 : pskb_expand_head+0x1/0x11b []
 0xc02ab5fe : ip_forward+0x150/0x256 []
 0xc02a9e9b : ip_rcv_finish+0xae/0xcb []
 0xc02aa314 : ip_rcv+0x207/0x231 []
 0xc031a0eb : notifier_call_chain+0x19/0x29 []
 0xc02902eb : netif_receive_skb+0x1e3/0x24e []
 0xc0291c60 : process_backlog+0x76/0xe3 []
 0xc0292048 : net_rx_action+0x94/0x14e []
 0xc0126e4d : __do_softirq+0x5d/0xba []
 0xc0126edc : do_softirq+0x32/0x36 []
 0xc01270a8 : local_bh_enable+0x7b/0x89 []
 0xc0292329 : dev_queue_xmit+0x227/0x246 []
 0xc031a0eb : notifier_call_chain+0x19/0x29 []
 0xc02aef41 : ip_output+0x1e6/0x221 []
 0xc02902eb : netif_receive_skb+0x1e3/0x24e []
 0xc02ae741 : ip_queue_xmit+0x3a8/0x3ea []
 0xc02c10cc : tcp_v4_send_check+0x74/0xaa []
 0xc02bc22a : tcp_transmit_skb+0x606/0x634 []
 0xc02aef41 : ip_output+0x1e6/0x221 []
 0xc02bdcec : tcp_push_one+0xb3/0xd8 []
 0xc02b429a : tcp_sendmsg+0x7c8/0x9f9 []
 0xc01cbe8a : vsnprintf+0x450/0x48c []
 0xc02cc10f : inet_sendmsg+0x3b/0x45 []
 0xc0286e17 : sock_sendmsg+0xd0/0xeb []
 0xc017be75 : seq_printf+0x2e/0x4b []
 0xc0133591 : autoremove_wake_function+0x0/0x35 []
 0xc0287691 : sys_sendto+0x116/0x140 []
 0xc02876f2 : sys_send+0x37/0x3b []
 0xc0288057 : sys_socketcall+0x14a/0x261 []
 0xc0102d6c : syscall_call+0x7/0xb []
```

Also here is a backtrace when etun_xmit() is called:

tx_dev: etun1
rx_dev: etun2

```
  0xf8b4a3b4 : etun_xmit+0x1/0xed [etun]
  0xc02906e0 : dev_hard_start_xmit+0x1ab/0x204 []
  0xc029229d : dev_queue_xmit+0x19b/0x246 []
  0xc0296672 : neigh_resolve_output+0x1f5/0x227 []
  0xc02aef41 : ip_output+0x1e6/0x221 []
  0xc02ab4ac : ip_forward_finish+0x2c/0x2e []
  0xc02ab6a5 : ip_forward+0x1f7/0x256 []           <- ip_forward
  0xc0292329 : dev_queue_xmit+0x227/0x246 []
  0xc02a9e9b : ip_rcv_finish+0xae/0xcb []
  0xc02aa314 : ip_rcv+0x207/0x231 []
  0xc028c0cd : __alloc_skb+0x4f/0xf8 []
  0xc02902eb : netif_receive_skb+0x1e3/0x24e []
  0xc0291c60 : process_backlog+0x76/0xe3 []
  0xc0292048 : net_rx_action+0x94/0x14e []
  0xc0126e4d : __do_softirq+0x5d/0xba []
  0xc0126edc : do_softirq+0x32/0x36 []
  0xc0104d2f : do_IRQ+0x87/0x9c []
  0xc0103707 : common_interrupt+0x23/0x28 []
  0xc0101c30 : default_idle+0x0/0x3e []
  0xc031007b : packet_set_ring+0x2d7/0x2ea []
  0xc0101c5c : default_idle+0x2c/0x3e []
  0xc0101390 : cpu_idle+0x9e/0xb7 []
```

That's it for now. I hope this helps.

Regards,
Benjamin

>
> Eric
> ____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers
>

--
B e n j a m i n   T h e r y - BULL/DT/Open Software R&D

   http://www.bull.com
____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers