Subject: [PATCHSET] 2.6.20-lxc8
Posted by Cedric Le Goater on Tue, 20 Mar 2007 20:53:01 GMT
View Forum Message <> Reply to Message

All,

We've been gathering, porting and testing a whole bunch of patchsets related to namespaces, containers and resource management in what we call the -lxc patchset.

The purpose of the -lxc patchset is to experiment and test a "fully" featured linux container solution. this is a large topic but resource management and container mobility are the main high level features we are looking at.

We're not there yet but we've done a lot of progress in 2006 and early 2007. It seems a good time to share some this work and see :

 * which patchsets can be proposed to andrew
 * which patchsets requires more work
 * which can be dropped
 * what kind of cleanups are still required
 * etc.

The latest release 2.6.20-lxc8 contains the following :

* user namespace from Serge E. Hallyn <serue@us.ibm.com>
* bind_ns() syscalls from Cedric Le Goater <clg@fr.ibm.com>
* ro bind mount from Dave Hansen <hansendc@us.ibm.com>
* generic Process containers from Paul Menage <menage@google.com>
* namespace entering from Serge E. Hallyn <serue@us.ibm.com>
* resource controllers based on process containers from Pavel Emelianov <xemul@sw.ru>
* multiple /proc (required for pid namespace) from Dave Hansen <hansendc@us.ibm.com>
* pid namespace from Sukadev Bhattiprolu <sukadev@us.ibm.com>
* L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
* misc fixes and cleanups from others (sorry for not mentioning)

and it's giving some good results on common platforms like i386 and x86_64.

It's all here :

 http://lxc.sourceforge.net/patches/2.6.20/2.6.20-lxc8

Some very basic user space tools to work and test namespaces can also be found here :

 http://lxc.sourceforge.net/patches/2.6.20/2.6.20-lxc8/broken-out/tests/

A lot of work still needs to be done in user space..this is not
the highest priority for the moment but sooner or later we'll
need to spend more time on it.

Hope you find it useful. If you have any issues with sourceforge
download, please tell me. I can move this patchset to a more
convenient site.

Cheers,

C.

____

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

---

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Herbert Poetzl on Wed, 21 Mar 2007 06:04:27 GMT
View Forum Message <> Reply to Message

On Tue, Mar 20, 2007 at 09:53:01PM +0100, Cedric Le Goater wrote:
> All,
>
> We've been gathering, porting and testing a whole bunch of patchsets
> related to namespaces, containers and resource management in what
> we call the -lxc patchset.

great!

> The purpose of the -lxc patchset is to experiment and test a "fully"
> featured linux container solution.

you mean, besides the existing "fully" featured linux container
solutions, like Linux-VServer, FreeVPS and OpenVZ

> this is a large topic but resource management and container mobility
> are the main high level features we are looking at.

okay, fair enough ...

> We're not there yet but we've done a lot of progress in 2006 and early
> 2007. It seems a good time to share some this work and see :
>
> * which patchsets can be proposed to andrew
> * which patchsets requires more work
> * which can be dropped

> * what kind of cleanups are still required
> * etc.
>
> The latest release 2.6.20-lxc8 contains the following :
>
> * user namespace from Serge E. Hallyn <serue@us.ibm.com>
> * bind_ns() syscalls from Cedric Le Goater <clg@fr.ibm.com>
> * ro bind mount from Dave Hansen <hansendc@us.ibm.com>

glad to read that this is also part of it ...

> * generic Process containers from Paul Menage <menage@google.com>
> * namespace entering from Serge E. Hallyn <serue@us.ibm.com>
> * resource controllers based on process containers from Pavel Emelianov <xemul@sw.ru>
> * multiple /proc (required for pid namespace) from Dave Hansen <hansendc@us.ibm.com>
> * pid namespace from Sukadev Bhattiprolu <sukadev@us.ibm.com>
> * L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
> * misc fixes and cleanups from others (sorry for not mentioning)
>
> and it's giving some good results on common platforms like i386 and
> x86_64.

what _are_ the good results? do you have performance
results or other interesting data on it? if so, where
can it be found?

TIA,
Herbert

> It's all here :
>
> http://lxc.sourceforge.net/patches/2.6.20/2.6.20-lxc8
>
> Some very basic user space tools to work and test namespaces can
> also be found here :
>
> http://lxc.sourceforge.net/patches/2.6.20/2.6.20-lxc8/broken-out/tests/
>
> A lot of work still needs to be done in user space..this is not
> the highest priority for the moment but sooner or later we'll
> need to spend more time on it.
>
> Hope you find it useful. If you have any issues with sourceforge
> download, please tell me. I can move this patchset to a more
> convenient site.
>
> Cheers,
>

> C.

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

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Dave Hansen on Wed, 21 Mar 2007 06:51:37 GMT
View Forum Message <> Reply to Message

On Wed, 2007-03-21 at 07:04 +0100, Herbert Poetzl wrote:
> > The purpose of the -lxc patchset is to experiment and test a "fully"
> > featured linux container solution.
>
> you mean, besides the existing "fully" featured linux container
> solutions, like Linux-VServer, FreeVPS and OpenVZ

The one difference here is that we're not generally planning on merging anything that's not mainline-bound. If you see stuff in there that you find reprehensible, please speak up, because it is probably headed to a mailing list near you with an [RFC] sometime in the near future. :)

Think of it like a little sandbox we can play with before things hit -mm.

-- Dave

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

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Daniel Lezcano on Wed, 21 Mar 2007 09:47:55 GMT
View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> On Tue, Mar 20, 2007 at 09:53:01PM +0100, Cedric Le Goater wrote:
>
>> All,
>>
>> We've been gathering, porting and testing a whole bunch of patchsets
>> related to namespaces, containers and resource management in what
>> we call the -lxc patchset.
>>

>
> great!
>
[cut]
>> * generic Process containers from Paul Menage <menage@google.com>
>> * namespace entering from Serge E. Hallyn <serue@us.ibm.com>
>> * resource controllers based on process containers from Pavel Emelianov <xemul@sw.ru>
>> * multiple /proc (required for pid namespace) from Dave Hansen <hansendc@us.ibm.com>
>> * pid namespace from Sukadev Bhattiprolu <sukadev@us.ibm.com>
>> * L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
>> * misc fixes and cleanups from others (sorry for not mentioning)
>>
>> and it's giving some good results on common platforms like i386 and
>> x86_64.
>>
>
> what _are_ the good results? do you have performance
> results or other interesting data on it? if so, where
> can it be found?
>
Hi Herbert,

I played with the L2 namespace patchset from Eric Biederman, I did some
benchmarking with netperf:

With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
Host A is running the netserver on a RH4 kernel 2.6.9-42
Host B is running the netperf client inside and outside the container
with the command:

```
 netperf -H HostA -c -l 20 -n 2 -p 12865
```

Results are:
inside the container:

```
   Throughput : 940.39  Mbit/s CPU usage : 15.80 %
```

outside the container:

```
   Throughput : 941.34 Mbits/s CPU usage :  5.80 %
```

I did the test again with 50 containers. I created them one by one
having one running netperf and the other being idle.
Each time I created a container, I rerun netperf. To be more explicit, I
created 1 container, run netperf inside it and blocked it on a fifo
reading, I created a second container, run netperf inside it and blocked
it, and son on ... to 50 containers. The benchmarking result are the
same as running one container, so I guess it scales well.

There are a lot of scenarii to do for benchmarking, for example, running
netperf in each container in the same time and look how it behaves.

I am profiling the kernel to look where the cpu overhead is.

Regards.

-- Daniel

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

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by dim on Wed, 21 Mar 2007 12:19:55 GMT

View Forum Message <> Reply to Message

On Wednesday 21 March 2007 12:47, Daniel Lezcano wrote:
> Herbert Poetzl wrote:
> > On Tue, Mar 20, 2007 at 09:53:01PM +0100, Cedric Le Goater wrote:
> >> All,
> >>
> >> We've been gathering, porting and testing a whole bunch of patchsets
> >> related to namespaces, containers and resource management in what
> >> we call the -lxc patchset.
> >
> > great!
>
> [cut]
>
> >> * generic Process containers from Paul Menage <menage@google.com>
> >> * namespace entering from Serge E. Hallyn <serue@us.ibm.com>
> >> * resource controllers based on process containers from Pavel Emelianov
> >> <xemul@sw.ru> * multiple /proc (required for pid namespace) from Dave
> >> Hansen <hansendc@us.ibm.com> * pid namespace from Sukadev Bhattiprolu
> >> <sukadev@us.ibm.com>
> >> * L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
> >> * misc fixes and cleanups from others (sorry for not mentioning)
> >>
> >> and it's giving some good results on common platforms like i386 and
> >> x86_64.
> >
> > what _are_ the good results? do you have performance
> > results or other interesting data on it? if so, where
> > can it be found?
>
> Hi Herbert,
>
> I played with the L2 namespace patchset from Eric Biederman, I did some
> benchmarking with netperf:

>
> With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
> Host A is running the netserver on a RH4 kernel 2.6.9-42
> Host B is running the netperf client inside and outside the container
> with the command:
> netperf -H HostA -c -l 20 -n 2 -p 12865
>
> Results are:
> inside the container:
>   Throughput : 940.39 Mbit/s CPU usage : 15.80 %
>
> outside the container:
>   Throughput : 941.34 Mbits/s CPU usage : 5.80 %

Daniel,

You probably did the same tests for my patchset also, didn't you? Which
results did you get?

>
> I did the test again with 50 containers. I created them one by one
> having one running netperf and the other being idle.
> Each time I created a container, I rerun netperf. To be more explicit, I
> created 1 container, run netperf inside it and blocked it on a fifo
> reading, I created a second container, run netperf inside it and blocked
> it, and son on ... to 50 containers. The benchmarking result are the
> same as running one container, so I guess it scales well.
>
> There are a lot of scenarii to do for benchmarking, for example, running
> netperf in each container in the same time and look how it behaves.
> I am profiling the kernel to look where the cpu overhead is.
>
> Regards.
>
>   -- Daniel

--
Thanks,
Dmitry.

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

---

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Daniel Lezcano on Wed, 21 Mar 2007 12:40:52 GMT
View Forum Message <> Reply to Message

Dmitry Mishin wrote:
> On Wednesday 21 March 2007 12:47, Daniel Lezcano wrote:
>
>> Herbert Poetzl wrote:
>>
>>> On Tue, Mar 20, 2007 at 09:53:01PM +0100, Cedric Le Goater wrote:
>>>
>>>> All,
>>>>
>>>> We've been gathering, porting and testing a whole bunch of patchsets
>>>> related to namespaces, containers and resource management in what
>>>> we call the -lxc patchset.
>>>>
>>> great!
>>>
>> [cut]
>>
>>
>>>> * generic Process containers from Paul Menage <menage@google.com>
>>>> * namespace entering from Serge E. Hallyn <serue@us.ibm.com>
>>>> * resource controllers based on process containers from Pavel Emelianov
>>>> <xemul@sw.ru> * multiple /proc (required for pid namespace) from Dave
>>>> Hansen <hansendc@us.ibm.com> * pid namespace from Sukadev Bhattiprolu
>>>> <sukadev@us.ibm.com>
>>>> * L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
>>>> * misc fixes and cleanups from others (sorry for not mentioning)
>>>>
>>>> and it's giving some good results on common platforms like i386 and
>>>> x86_64.
>>>>
>>> what _are_ the good results? do you have performance
>>> results or other interesting data on it? if so, where
>>> can it be found?
>>>
>> Hi Herbert,
>>
>> I played with the L2 namespace patchset from Eric Biederman, I did some
>> benchmarking with netperf:
>>
>> With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
>> Host A is running the netserver on a RH4 kernel 2.6.9-42
>> Host B is running the netperf client inside and outside the container
>> with the command:
>> netperf -H HostA -c -l 20 -n 2 -p 12865
>>
>> Results are:
>> inside the container:
>>   Throughput : 940.39 Mbit/s CPU usage : 15.80 %

>>
>> outside the container:
>>   Throughput : 941.34 Mbits/s CPU usage : 5.80 %
>>
> Daniel,
>
> You probably did the same tests for my patchset also, didn't you? Which
> results did you get?
>
Effectively, I did some tests with your patchset but in a different way.
I did it with a bridge and with tbench so I didn't got the cpu usage and
througput is impacted. It will be irrelevant to give these values if we
can not compare them with Eric's patchset.
Anyway, you are right, it is interesting to have a comparison. For this
reason I added the ioctl in veth to facilitate automated benchmarking
and I am finishing the performances test suite for your patchset. I
will send the results ASAP.

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

---

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Herbert Poetzl on Wed, 21 Mar 2007 13:51:23 GMT

View Forum Message <> Reply to Message

On Wed, Mar 21, 2007 at 01:40:52PM +0100, Daniel Lezcano wrote:
> Dmitry Mishin wrote:
> >On Wednesday 21 March 2007 12:47, Daniel Lezcano wrote:
> >
> >>Herbert Poetzl wrote:
> >>
> >>>On Tue, Mar 20, 2007 at 09:53:01PM +0100, Cedric Le Goater wrote:
> >>>
> >>>>All,
> >>>>
> >>>>We've been gathering, porting and testing a whole bunch of patchsets
> >>>>related to namespaces, containers and resource management in what
> >>>>we call the -lxc patchset.
> >>>>
> >>>great!
> >>>
> >>[cut]
> >>
> >>
> >>>>* generic Process containers from Paul Menage <menage@google.com>

> >>>>* namespace entering from Serge E. Hallyn <serue@us.ibm.com>
> >>>>* resource controllers based on process containers from Pavel Emelianov
> >>>><xemul@sw.ru> * multiple /proc (required for pid namespace) from Dave
> >>>>Hansen <hansendc@us.ibm.com> * pid namespace from Sukadev Bhattiprolu
> >>>><sukadev@us.ibm.com>
> >>>>* L2 network namespace from Eric W. Biederman <ebiederm@xmission.com>
> >>>>* misc fixes and cleanups from others (sorry for not mentioning)
> >>>>
> >>>>and it's giving some good results on common platforms like i386 and
> >>>>x86_64.
> >>>>
> >>>what _are_ the good results? do you have performance
> >>>results or other interesting data on it? if so, where
> >>>can it be found?
> >>>
> >>Hi Herbert,
> >>
> >>I played with the L2 namespace patchset from Eric Biederman, I did some
> >>benchmarking with netperf:
> >>
> >>With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
> >>Host A is running the netserver on a RH4 kernel 2.6.9-42
> >>Host B is running the netperf client inside and outside the container
> >>with the command:
> >> netperf -H HostA -c -l 20 -n 2 -p 12865
> >>
> >>Results are:
> >>inside the container:
> >>   Throughput : 940.39 Mbit/s CPU usage : 15.80 %
> >>
> >>outside the container:
> >>   Throughput : 941.34 Mbits/s CPU usage : 5.80 %
> >>
> >Daniel,
> >
> >You probably did the same tests for my patchset also, didn't you?
> >Which results did you get?
> >
> Effectively, I did some tests with your patchset but in a different way.
> I did it with a bridge and with tbench so I didn't got the cpu usage and
> througput is impacted. It will be irrelevant to give these values if we
> can not compare them with Eric's patchset.
> Anyway, you are right, it is interesting to have a comparison. For this
> reason I added the ioctl in veth to facilitate automated benchmarking
> and I am finishing the performances test suite for your patchset. I
> will send the results ASAP.

excellent, please don't forget to test with L3 isoltaion

(a simple network namespace from Linux-VServer should
do the trick) too, let me know if you have any issues
with getting started ...

TIA,
Herbert

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

---

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Wed, 21 Mar 2007 18:12:45 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Hi Herbert,
>
> I played with the L2 namespace patchset from Eric Biederman, I did some
> benchmarking with netperf:
>
> With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
> Host A is running the netserver on a RH4 kernel 2.6.9-42
> Host B is running the netperf client inside and outside the container with the
> command:
> netperf -H HostA -c -l 20 -n 2 -p 12865
>
> Results are:
> inside the container:
>   Throughput : 940.39 Mbit/s CPU usage : 15.80 %
>
> outside the container:
>   Throughput : 941.34 Mbits/s CPU usage : 5.80 %

Could you clarify a little bit what you mean by inside the container
and outside the container?   In particular was the outside the container case
in a kernel that had the network namespace compiled in?

Could you also clarify how you have setup networking inside the container
and going to the outside world?

> I did the test again with 50 containers. I created them one by one having one
> running netperf and the other being idle.
> Each time I created a container, I rerun netperf. To be more explicit, I created
> 1 container, run netperf inside it and blocked it on a fifo reading, I created a

> second container, run netperf inside it and blocked it, and son on ... to 50
> containers. The benchmarking result are the same as running one container, so I
> guess it scales well.
>
> There are a lot of scenarii to do for benchmarking, for example, running netperf
> in each container in the same time and look how it behaves.
> I am profiling the kernel to look where the cpu overhead is.

Thanks. This is simple enough that at least part of this should be easy
to reproduce. Once things settle down a little I'm interested in tracking
the cpu usage if someone else hasn't done it already.

Eric

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

---

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by Daniel Lezcano on Wed, 21 Mar 2007 23:04:07 GMT

View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
>
>
>> Hi Herbert,
>>
>> I played with the L2 namespace patchset from Eric Biederman, I did some
>> benchmarking with netperf:
>>
>> With 2 hosts, Intel EM64T bipro HT / 2,4 GHz , 4Go ram and GB network.
>> Host A is running the netserver on a RH4 kernel 2.6.9-42
>> Host B is running the netperf client inside and outside the container with the
>> command:
>> netperf -H HostA -c -l 20 -n 2 -p 12865
>>
>> Results are:
>> inside the container:
>>   Throughput : 940.39 Mbit/s CPU usage : 15.80 %
>>
>> outside the container:
>>   Throughput : 941.34 Mbits/s CPU usage : 5.80 %
>>
>
> Could you clarify a little bit what you mean by inside the container
> and outside the container?   In particular was the outside the container case

> in a kernel that had the network namespace compiled in?
>
Sure.
Outside the container means : no container and netperf is ran into a kernel with the network namespace compiled in.
Inside the container means : a network namespace is created with the configuration described below and netperf is ran inside this network namespace.
I ran a netperf on a vanilla kernel too and netperf results are roughly the same than netperf ran on a network namespace kernel.

Anyway, I want to do some benchmarking on Dmitry's patchset and Linux-Verser and I will come back with a more complete benchmarking results in a few days.

> Could you also clarify how you have setup networking inside the container
> and going to the outside world?
>
Ok. Let's assume we have host A which is a RH4 kernel with netperf server running and with an IP address 1.2.3.100
Host B is the kernel with network namespace compiled in. The objective is to be able to setup the container without doing extra network configuration on host A.
The physical network interface is eth0, we want to have IPs for etun pair-device 1.2.3.4/1.2.3.5, we have 2 shells A and B
So I did the following:

```
   (in shell A)
   # do proxy arp for eth0
   echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp

   # add in the init namespace IP address of the same network than the
containers
   # it is more comfortable. We can add a route -net otherwise
   ifconfig eth0:0 1.2.3.1

   # create the pair device
   echo etun1,etun2 > /sys/module/etun/parameters/newif

   # do proxy arp for etun1
   echo 1 > /proc/sys/net/ipv4/conf/etun1/proxy_arp

   # set ip address for etun1
   ifconfig etun1 1.2.3.4

   # route packet for etun2 IP to etun1
   route add -host 1.2.3.5 dev etun1
```

```
    (in shell B)
    # unshare network namespace

    (in shell A)
    # move etun2 to network namespace
    echo pid_of_shell_B > /sys/class/net/etun2/new_ns_pid

    (in shell B)
    # set the etun2 IP address
    ifconfig etun2 1.2.3.5

    # set the loopback up
   ifconfig lo up

    # check ping, 1.2.3.4, 1.2.3.5, 1.2.3.1, 1.2.3.100 is successful
    # run netperf or what ever you want
```

>> I did the test again with 50 containers. I created them one by one having one
>> running netperf and the other being idle.
>> Each time I created a container, I rerun netperf. To be more explicit, I created
>> 1 container, run netperf inside it and blocked it on a fifo reading, I created a
>> second container, run netperf inside it and blocked it, and son on ... to 50
>> containers. The benchmarking result are the same as running one container, so I
>> guess it scales well.
>>
>> There are a lot of scenarii to do for benchmarking, for example, running netperf
>> in each container in the same time and look how it behaves.
>> I am profiling the kernel to look where the cpu overhead is.
>>
>
> Thanks. This is simple enough that at least part of this should be easy
> to reproduce. Once things settle down a little I'm interested in tracking
> the cpu usage if someone else hasn't done it already.
>

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.

Regards
   -- Daniel

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

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Thu, 22 Mar 2007 06:30:18 GMT

View Forum Message <> Reply to Message

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?

Eric

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

Subject: Re: [PATCHSET] 2.6.20-lxc8
Posted by dev on Thu, 22 Mar 2007 09:38:25 GMT

View Forum Message <> Reply to Message

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.

Thanks,
Kirill

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

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Thu, 22 Mar 2007 20:02:38 GMT

View Forum Message <> Reply to Message

Benjamin Thery <benjamin.thery@bull.net> writes:

> 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).

Ok. This seems to make sense, and is related to how we have configured the
network in this case.

It looks like pskb_expand_head is called by skb_cow.

skb_cow has two cases when it calls pskb_expand_head.

- When there are multiple people who have a copy of the packet
  (tcpdump and friends)
- When there isn't enough room for the hard header.

Any chance one of you guys looking into this can instrument up
ip_foward just before the call to skb_cow and find out which
reason it is?

A cheap trick to make the overhead go away is probably to setup
ethernet bridging in this case...

But if we can ensure the ip_foward case does not need to do anything
more than modify the ttl and update the destination that would
be good to.

Anyway this does look very solvable.

Eric

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

Subject: R
Posted by Benjamin Thery on Thu, 22 Mar 2007 20:18:51 GMT
View Forum Message <> Reply to Message

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

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by Kirill Korotaev on Fri, 23 Mar 2007 09:35:01 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Benjamin Thery <benjamin.thery@bull.net> writes:
>
>
>>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).
>
>
> Ok. This seems to make sense, and is related to how we have configured the
> network in this case.
>
> It looks like pskb_expand_head is called by skb_cow.
>
> skb_cow has two cases when it calls pskb_expand_head.
> - When there are multiple people who have a copy of the packet
>   (tcpdump and friends)
> - When there isn't enough room for the hard header.
>
> Any chance one of you guys looking into this can instrument up
> ip_foward just before the call to skb_cow and find out which
> reason it is?
>
> A cheap trick to make the overhead go away is probably to setup
> ethernet bridging in this case...
>
> But if we can ensure the ip_foward case does not need to do anything
> more than modify the ttl and update the destination that would
> be good to.
>
> Anyway this does look very solvable.

we have the hack below in ip_forward() to avoid skb_cow(),
Banjamin, can you check whether it helps in your case please?
(NOTE: you will need to replace check for NETIF_F_VENET with something else
 or introduce the same flag on etun device).

diff -upr linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c
linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c

```
--- linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c	2006-09-20 07:42:06.000000000 +0400
+++ linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c 2007-03-20 17:22:45.000000000 +0300
@@ -86,6 +86,24 @@ int ip_forward(struct sk_buff *skb)
 	if (opt->is_strictroute && rt->rt_dst != rt->rt_gateway)
 		goto sr_failed;
 
+	/*
+	 * We try to optimize forwarding of VE packets:
+	 * do not decrement TTL (and so save skb_cow)
+	 * during forwarding of outgoing pkts from VE.
+	 * For incoming pkts we still do ttl decr,
+	 * since such skb is not cloned and does not require
+	 * actual cow. So, there is at least one place
+	 * in pkts path with mandatory ttl decr, that is
+	 * sufficient to prevent routing loops.
+	 */
+	iph = skb->nh.iph;
+	if (
+#ifdef CONFIG_IP_ROUTE_NAT
+	    (rt->rt_flags & RTCF_NAT) == 0 &&	  /* no NAT mangling expected */
+#endif					  /* and */
+	    (skb->dev->features & NETIF_F_VENET)) /* src is VENET device */
+		goto no_ttl_decr;
+
 	/* We are about to mangle packet. Copy it! */
 	if (skb_cow(skb, LL_RESERVED_SPACE(rt->u.dst.dev)+rt->u.dst.header_len))
 		goto drop;
@@ -94,6 +112,8 @@ int ip_forward(struct sk_buff *skb)
 	/* Decrease ttl after skb cow done */
 	ip_decrease_ttl(iph);
 
+no_ttl_decr:
+
 	/*
 	 *	We now generate an ICMP HOST REDIRECT giving the route
 	 *	we calculated.
@@ -121,3 +141,5 @@ drop:
```

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

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Fri, 23 Mar 2007 10:34:01 GMT

View Forum Message <> Reply to Message

Kirill Korotaev <dev@openvz.org> writes:

> we have the hack below in ip_forward() to avoid skb_cow(),
> Banjamin, can you check whether it helps in your case please?
> (NOTE: you will need to replace check for NETIF_F_VENET with something else
> or introduce the same flag on etun device).

Ugh. The thing is skb_cow should be free. It only has a cost when the skb
is too small or there is a second copy of the skb. I don't there is a technical
reason for either of those to be the case when we are going over ethernet.

And since the hardware header needs to change as well your hack is actually broken
if the incoming network interface is not ethernet.

So while I can see this hack for testing I'd much rather see if we can actually
fix this one cleanly.

Unless you understand what is triggering the skb_cow to actually perform
the copy.

Eric

```
> diff -upr linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c
> linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c
> --- linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c 2006-09-20 07:42:06.000000000
> +0400
> +++ linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c 2007-03-20
> 17:22:45.000000000 +0300
> @@ -86,6 +86,24 @@ int ip_forward(struct sk_buff *skb)
>  	if (opt->is_strictroute && rt->rt_dst != rt->rt_gateway)
>  		goto sr_failed;
> 
> +	/*
> +	 * We try to optimize forwarding of VE packets:
> +	 * do not decrement TTL (and so save skb_cow)
> +	 * during forwarding of outgoing pkts from VE.
> +	 * For incoming pkts we still do ttl decr,
> +	 * since such skb is not cloned and does not require
> +	 * actual cow. So, there is at least one place
> +	 * in pkts path with mandatory ttl decr, that is
> +	 * sufficient to prevent routing loops.
> +	 */
> +	iph = skb->nh.iph;
> +	if (
> +#ifdef CONFIG_IP_ROUTE_NAT
> + (rt->rt_flags & RTCF_NAT) == 0 && /* no NAT mangling expected */
> +#endif					  /* and */
> +	    (skb->dev->features & NETIF_F_VENET)) /* src is VENET device */
> +		goto no_ttl_decr;
> +
>         /* We are about to mangle packet. Copy it! */
>         if (skb_cow(skb, LL_RESERVED_SPACE(rt->u.dst.dev)+rt->u.dst.header_len))
>                 goto drop;
> @@ -94,6 +112,8 @@ int ip_forward(struct sk_buff *skb)
>         /* Decrease ttl after skb cow done */
>         ip_decrease_ttl(iph);
>
> +no_ttl_decr:
> +
>         /*
>          *      We now generate an ICMP HOST REDIRECT giving the route
>          *      we calculated.
> @@ -121,3 +141,5 @@ drop:
```

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by Benjamin Thery on Tue, 27 Mar 2007 16:34:44 GMT

View Forum Message <> Reply to Message

Hi,

Yesterday, I applied a patch similar to Kirill's one that skip
skb_cow() in ip_forward when the device is a etun, and it does help a lot.

With the patch the cpu load increase is reduced by 50%. Part of the
problem is "solved".

Here are the figures for netperf:

(Host A -> Host B
  Host A is running kernel 2.6.20-rc5-netns.i386)

```
                              Throughput    CPU load

- without container:               719.78     10.45
- inside a container (no patch)   719.37     21.88
- inside a container with patch:  728.93     15.41
```

The CPU load with the ip_forward patch is now "only" 50% higher (10%
compared to 15%) than the reference case without container.

The throughput is even better (I repeated the test a few times and I
always got better results from inside the container).

(1) Why skb_cow() performs the copy?

I also added some traces to understand why skb_cow() does copy the
skb: is it insufficient headroom or that the skb has been cloned
previously?
In our case, the condition is always that the "TCP skb" is marked as
cloned.
It is likely that these skb have been cloned in tcp_skb_transmit().

(2) Who consumes the other 5% percent cpu?

With the patch installed oprofile reports that pskb_expand_head()
(called by skb_cow) has disappeared from the top cpu consumers list.

Now, the remaining symbol that shows unusual activity is
csum_partial_copy_generic().
I'd like to find who is the caller, unfortunately, this one is harder
to track. It is written in assembler and called by "static inline"
routines and Systemtap doesn't like that. :(

So, that was the current status.
I'm continuing my investigations.

Regards,
Benjamin

Eric W. Biederman wrote:
> Kirill Korotaev <dev@openvz.org> writes:
>
>> we have the hack below in ip_forward() to avoid skb_cow(),
>> Banjamin, can you check whether it helps in your case please?
>> (NOTE: you will need to replace check for NETIF_F_VENET with something else
>> or introduce the same flag on etun device).
>
> Ugh. The thing is skb_cow should be free. It only has a cost when the skb
> is too small or there is a second copy of the skb. I don't there is a technical
> reason for either of those to be the case when we are going over ethernet.
>
> And since the hardware header needs to change as well your hack is actually broken
> if the incoming network interface is not ethernet.
>
> So while I can see this hack for testing I'd much rather see if we can actually
> fix this one cleanly.
>
> Unless you understand what is triggering the skb_cow to actually perform

```
> the copy.
>
> Eric
>
>> diff -upr linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c
>> linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c
>> --- linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c 2006-09-20 07:42:06.000000000
>> +0400
>> +++ linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c 2007-03-20
>> 17:22:45.000000000 +0300
>> @@ -86,6 +86,24 @@ int ip_forward(struct sk_buff *skb)
>>       if (opt->is_strictroute && rt->rt_dst != rt->rt_gateway)
>>               goto sr_failed;
>>
>> +       /*
>> +        * We try to optimize forwarding of VE packets:
>> +        * do not decrement TTL (and so save skb_cow)
>> +        * during forwarding of outgoing pkts from VE.
>> +        * For incoming pkts we still do ttl decr,
>> +        * since such skb is not cloned and does not require
>> +        * actual cow. So, there is at least one place
>> +        * in pkts path with mandatory ttl decr, that is
>> +        * sufficient to prevent routing loops.
>> +        */
>> +       iph = skb->nh.iph;
>> +       if (
>> +#ifdef CONFIG_IP_ROUTE_NAT
>> + (rt->rt_flags & RTCF_NAT) == 0 && /* no NAT mangling expected */
>> +#endif                                          /* and */
>> +           (skb->dev->features & NETIF_F_VENET)) /* src is VENET device */
>> +               goto no_ttl_decr;
>> +
>>       /* We are about to mangle packet. Copy it! */
>>       if (skb_cow(skb, LL_RESERVED_SPACE(rt->u.dst.dev)+rt->u.dst.header_len))
>>               goto drop;
>> @@ -94,6 +112,8 @@ int ip_forward(struct sk_buff *skb)
>>       /* Decrease ttl after skb cow done */
>>       ip_decrease_ttl(iph);
>>
>> +no_ttl_decr:
>> +
>>       /*
>>        *      We now generate an ICMP HOST REDIRECT giving the route
>>        *      we calculated.
>> @@ -121,3 +141,5 @@ drop:
>
```

--
Benjamin Thery - BULL/DT/Open Software R&D

   http://www.bull.com

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Wed, 28 Mar 2007 00:31:59 GMT

View Forum Message <> Reply to Message

Benjamin Thery <benjamin.thery@bull.net> writes:

```
> Hi,
>
> Yesterday, I applied a patch similar to Kirill's one that skip skb_cow() in
> ip_forward when the device is a etun, and it does help a lot.
>
> With the patch the cpu load increase is reduced by 50%. Part of the problem is
> "solved".

>
> Here are the figures for netperf:
>
> (Host A -> Host B
>  Host A is running kernel 2.6.20-rc5-netns.i386)
>
>                          Throughput    CPU load
>
> - without container:              719.78     10.45
> - inside a container (no patch)   719.37     21.88
> - inside a container with patch:  728.93     15.41
>
> The CPU load with the ip_forward patch is now "only" 50% higher (10% compared to
> 15%) than the reference case without container.
>
> The throughput is even better (I repeated the test a few times and I always got
> better results from inside the container).
>
> (1) Why skb_cow() performs the copy?
>
> I also added some traces to understand why skb_cow() does copy the skb: is it
> insufficient headroom or that the skb has been cloned previously?
> In our case, the condition is always that the "TCP skb" is marked as cloned.
> It is likely that these skb have been cloned in tcp_skb_transmit().
```

Hmm. I wonder if there is any way we could possibly detect or avoid that case.
It sounds like a general routing code issue if the copy is unnecessary.

> (2) Who consumes the other 5% percent cpu?
>
> With the patch installed oprofile reports that pskb_expand_head() (called by
> skb_cow) has disappeared from the top cpu consumers list.
>
> Now, the remaining symbol that shows unusual activity is
> csum_partial_copy_generic().
> I'd like to find who is the caller, unfortunately, this one is harder to
> track. It is written in assembler and called by "static inline" routines and
> Systemtap doesn't like that. :(
>
>
> So, that was the current status.
> I'm continuing my investigations.

Thanks. I would recommend testing a setup using the in kernel ethernet bridging.
It is a completely different path and it should not have much less of a potential
to process packets before they get to the destination network namespace.

Of course if we can improve our routing performance that would be good but
there are limits to what we can correctly do.

Eric

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by dev on Wed, 28 Mar 2007 08:01:41 GMT

View Forum Message <> Reply to Message

Benjamin,

checksumming can be optimized out as well.
We had an experimental patch for OpenVZ venet device, which adds
NETIF_F_LLTX | NETIF_F_HW_CSUM | NETIF_F_SG | NETIF_F_HIGHDMA
features to venet device and avoids additional checksumming where possible
(moving RX/TX checksum calculation to hardware).

So I guess this is doable in future as well.

Thanks,
Kirill

> Hi,
>
> Yesterday, I applied a patch similar to Kirill's one that skip
> skb_cow() in ip_forward when the device is a etun, and it does help a lot.
>
> With the patch the cpu load increase is reduced by 50%. Part of the
> problem is "solved".
>
> Here are the figures for netperf:
>
> (Host A -> Host B
>   Host A is running kernel 2.6.20-rc5-netns.i386)
>
```
>                             Throughput    CPU load
>
> - without container:              719.78     10.45
> - inside a container (no patch)   719.37     21.88
> - inside a container with patch:  728.93     15.41
```
>
> The CPU load with the ip_forward patch is now "only" 50% higher (10%
> compared to 15%) than the reference case without container.
>
> The throughput is even better (I repeated the test a few times and I
> always got better results from inside the container).
>
> (1) Why skb_cow() performs the copy?
>
> I also added some traces to understand why skb_cow() does copy the
> skb: is it insufficient headroom or that the skb has been cloned
> previously?
> In our case, the condition is always that the "TCP skb" is marked as
> cloned.
> It is likely that these skb have been cloned in tcp_skb_transmit().
>
>
> (2) Who consumes the other 5% percent cpu?
>
> With the patch installed oprofile reports that pskb_expand_head()
> (called by skb_cow) has disappeared from the top cpu consumers list.
>
> Now, the remaining symbol that shows unusual activity is
> csum_partial_copy_generic().
> I'd like to find who is the caller, unfortunately, this one is harder
> to track. It is written in assembler and called by "static inline"
> routines and Systemtap doesn't like that. :(

>
>
> So, that was the current status.
> I'm continuing my investigations.
>
> Regards,
> Benjamin
>
> Eric W. Biederman wrote:
>
>>Kirill Korotaev <dev@openvz.org> writes:
>>
>>
>>>we have the hack below in ip_forward() to avoid skb_cow(),
>>>Banjamin, can you check whether it helps in your case please?
>>>(NOTE: you will need to replace check for NETIF_F_VENET with something else
>>> or introduce the same flag on etun device).
>>
>>Ugh. The thing is skb_cow should be free. It only has a cost when the skb
>>is too small or there is a second copy of the skb. I don't there is a technical
>>reason for either of those to be the case when we are going over ethernet.
>>
>>And since the hardware header needs to change as well your hack is actually broken
>>if the incoming network interface is not ethernet.
>>
>>So while I can see this hack for testing I'd much rather see if we can actually
>>fix this one cleanly.
>>
>>Unless you understand what is triggering the skb_cow to actually perform
>>the copy.
>>
>>Eric
>>
>>

```
>>>diff -upr linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c
>>>linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c
>>>--- linux-2.6.18-rhel5.orig/net/ipv4/ip_forward.c 2006-09-20 07:42:06.000000000
>>>+0400
>>>+++ linux-2.6.18-rhel5-028stab023/net/ipv4/ip_forward.c 2007-03-20
>>>17:22:45.000000000 +0300
>>>@@ -86,6 +86,24 @@ int ip_forward(struct sk_buff *skb)
>>>        if (opt->is_strictroute && rt->rt_dst != rt->rt_gateway)
>>>                goto sr_failed;
>>>
>>>+       /*
>>>+        * We try to optimize forwarding of VE packets:
>>>+        * do not decrement TTL (and so save skb_cow)
>>>+        * during forwarding of outgoing pkts from VE.
```

```
>>>+        * For incoming pkts we still do ttl decr,
>>>+        * since such skb is not cloned and does not require
>>>+        * actual cow. So, there is at least one place
>>>+        * in pkts path with mandatory ttl decr, that is
>>>+        * sufficient to prevent routing loops.
>>>+        */
>>>+       iph = skb->nh.iph;
>>>+       if (
>>>+#ifdef CONFIG_IP_ROUTE_NAT
>>>+ (rt->rt_flags & RTCF_NAT) == 0 && /* no NAT mangling expected */
>>>+#endif                                        /* and */
>>>+           (skb->dev->features & NETIF_F_VENET)) /* src is VENET device */
>>>+               goto no_ttl_decr;
>>>+
>>>        /* We are about to mangle packet. Copy it! */
>>>        if (skb_cow(skb, LL_RESERVED_SPACE(rt->u.dst.dev)+rt->u.dst.header_len))
>>>                goto drop;
>>>@@ -94,6 +112,8 @@ int ip_forward(struct sk_buff *skb)
>>>        /* Decrease ttl after skb cow done */
>>>        ip_decrease_ttl(iph);
>>>
>>>+no_ttl_decr:
>>>+
>>>        /*
>>>         *      We now generate an ICMP HOST REDIRECT giving the route
>>>         *      we calculated.
>>>@@ -121,3 +141,5 @@ drop:
>>
>
>
```

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

---

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Wed, 28 Mar 2007 12:09:22 GMT

View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Benjamin,
>
> checksumming can be optimized out as well.
> We had an experimental patch for OpenVZ venet device, which adds
> NETIF_F_LLTX | NETIF_F_HW_CSUM | NETIF_F_SG | NETIF_F_HIGHDMA

> features to venet device and avoids additional checksumming where possible
> (moving RX/TX checksum calculation to hardware).
>
> So I guess this is doable in future as well.

I think I have the checksum bits settable in software with etun already. If not
it shouldn't be to hard to add.

I don't default to that because depending on your configuration it might not
be safe. In particular I think when you are using ethernet bridging we
need to do the packet checksum immediately off the wire.

Eric

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

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by Benjamin Thery on Wed, 28 Mar 2007 12:30:26 GMT

View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Kirill Korotaev <dev@sw.ru> writes:
>
>> Benjamin,
>>
>> checksumming can be optimized out as well.
>> We had an experimental patch for OpenVZ venet device, which adds
>> NETIF_F_LLTX | NETIF_F_HW_CSUM | NETIF_F_SG | NETIF_F_HIGHDMA
>> features to venet device and avoids additional checksumming where possible
>> (moving RX/TX checksum calculation to hardware).
>>
>> So I guess this is doable in future as well.
>
> I think I have the checksum bits settable in software with etun already. If not
> it shouldn't be to hard to add.

I tried to activate the checksum offload on etun to see if it improves
things, but, unfortunately once activated all my traffic was lost or
blocked. I didn't spend a lot of time on the issue. May be I'll give
it another try.

BTW, there is a small bug in etun_set_tx_csum. We can't disable the
checksum offloading once it has been set. I think it should look like
this: (sorry I haven't a patch ready)

```
static int etun_set_tx_csum(struct net_device *dev, u32 data)
{
 if (data)
  dev->features |= NETIF_F_NO_CSUM;
 else
  dev->features &= ~NETIF_F_NO_CSUM;
 return 0;
}
```

>
> I don't default to that because depending on your configuration it might not
> be safe. In particular I think when you are using ethernet bridging we
> need to do the packet checksum immediately off the wire.
>
> Eric
>

--
Benjamin Thery - 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

Subject: Re: Re: [PATCHSET] 2.6.20-lxc8
Posted by ebiederm on Wed, 28 Mar 2007 13:24:07 GMT

View Forum Message <> Reply to Message

Benjamin Thery <benjamin.thery@bull.net> writes:

> I tried to activate the checksum offload on etun to see if it improves things,
> but, unfortunately once activated all my traffic was lost or blocked. I didn't
> spend a lot of time on the issue. May be I'll give it another try.

No problem.

I don't much care right now once we have isolated the problem down to the
extra hops needed to reach the outside world.

> BTW, there is a small bug in etun_set_tx_csum. We can't disable the checksum
> offloading once it has been set. I think it should look like this: (sorry I
> haven't a patch ready)
>

```
> static int etun_set_tx_csum(struct net_device *dev, u32 data)
> {
>  if (data)
>   dev->features |= NETIF_F_NO_CSUM;
>  else
>   dev->features &= ~NETIF_F_NO_CSUM;
>  return 0;
> }
```

Looking at the code I would actually say looking like:

```
static int etun_set_tx_csum(struct net_device *dev, u32 data)
{
 dev->features &= ~NETIF_F_NO_CSUM;
 if (data)
  dev->features |= NETIF_F_NO_CSUM;

 return 0;
}
```

Is what I intended. It seems I forgot the ~.....
Oops.

Eric