Subject: Problem with bonding, vlan, bridge, veth
Posted by kfh on Fri, 10 Nov 2006 10:12:55 GMT
View Forum Message <> Reply to Message

Hi list,

I have a bonding/vlan/bridge/veth problem.
Sometimes a bridge think a veth device move to another port.
If I remove a physical interface from bond, the bridge behaves normally.

Kernel 2.6.16 + openvz test020
VE0 Ubuntu dapper/6.06LTS, IP 172.31.1.26 on VLAN 254
VE1028 Debian stable/sarge/3.1, IP 10.1.28.12 on VLAN 28

I have a server (vs5, VE0) using eth0 and eth1 in a bonding interface bond0.
bond0 is on tagged vlan.
I create a vlan device vlan254 on vlan 254. This is VE0 IP.
For each VE (XX) I do
 create a vlan device vlanXX on vlan XX.
 create a bridge bvXX and add vlanXX to it.
 create a VE (VE10XX) using veth.
 VETH="ve10XX.0,aa:00:04:56:YY:ZZ,eth0,aa:00:04:57:YY:ZZ"
 add ve10XX.0 to the bridge.
 YY and ZZ are calculated from VEID number (VLAN + 1000)

```
    eth0     eth1
      \     /
       bond0
      /     \                veth
 vlan254      vlanXX    ve10XX.0 -- eth0 (ve10XX)
  VE0           \    /
                bvXX (bridge)
```

I create and start VE1028, now I have:

```
VE0# ifconfig
bond0     Link encap:Ethernet  HWaddr 00:18:8B:2F:5F:F2
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:888940 errors:0 dropped:0 overruns:0 frame:0
          TX packets:150577 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:71916311 (68.5 MiB)  TX bytes:27093123 (25.8 MiB)

bv28      Link encap:Ethernet  HWaddr 00:18:8B:2F:5F:F2
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

RX bytes:212782 (207.7 KiB) TX bytes:0 (0.0 b)

eth0 Link encap:Ethernet HWaddr 00:18:8B:2F:5F:F2
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:659778 errors:0 dropped:0 overruns:0 frame:0
TX packets:150577 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:56333295 (53.7 MiB) TX bytes:27093123 (25.8 MiB)
Base address:0xecc0 Memory:dfae0000-dfb00000

eth1 Link encap:Ethernet HWaddr 00:18:8B:2F:5F:F2
UP BROADCAST RUNNING NOARP SLAVE MULTICAST MTU:1500 Metric:1
RX packets:229162 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:15583016 (14.8 MiB) TX bytes:0 (0.0 b)
Base address:0xdcc0 Memory:df8e0000-df900000

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:352 (352.0 b) TX bytes:352 (352.0 b)

ve1028.0 Link encap:Ethernet HWaddr AA:00:04:56:04:04
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:225 errors:0 dropped:0 overruns:0 frame:0
TX packets:4700 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:41399 (40.4 KiB) TX bytes:260688 (254.5 KiB)

venet0 Link encap:UNSPEC HWaddr
00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
UP BROADCAST POINTOPOINT RUNNING NOARP MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

vlan28 Link encap:Ethernet HWaddr 00:18:8B:2F:5F:F2
UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
RX packets:190890 errors:0 dropped:0 overruns:0 frame:0
TX packets:32868 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:11008978 (10.4 MiB) TX bytes:4038500 (3.8 MiB)

```
vlan254   Link encap:Ethernet  HWaddr 00:18:8B:2F:5F:F2
          inet addr:172.31.1.26  Bcast:172.31.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:490936 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77435 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23453611 (22.3 MiB)  TX bytes:10026463 (9.5 MiB)


VE1028# ifconfig
eth0      Link encap:Ethernet  HWaddr AA:00:04:57:04:04
          inet addr:10.1.28.12  Bcast:10.1.28.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4887 errors:0 dropped:0 overruns:0 frame:0
          TX packets:231 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:271148 (264.7 KiB)  TX bytes:43395 (42.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)


>From VE1028 I ping a router (10.1.28.4)
VE1028# ping 10.1.28.4

VE0# brctl showmacs bv28
port no mac addr                is local?       ageing timer
  1     00:18:8b:2f:5f:f2       yes             0.00
  1     02:e0:52:16:95:1c       no              0.00
  2     aa:00:04:56:04:04       yes             0.00
  2     aa:00:04:57:04:04       no              0.00

>From VE1028 I ping another router (10.1.28.101)
I don't get arp replies in VE1028
If I run tcpdump on VE0/bv28, I see the replies.

VE0# brctl showmacs bv28
port no mac addr                is local?       ageing timer
  1     00:03:fa:0f:a3:a7       no              0.15
  1     00:18:8b:2f:5f:f2       yes             0.00
  1     02:e0:52:16:95:1c       no              0.79
  2     aa:00:04:56:04:04       yes             0.00
  1     aa:00:04:57:04:04       no              0.15
```

Now the bridge thinks VE1028/eth0 moved to port 1.
aa:00:04:57:04:04 never gets the replies, as the bridge
doesn't forward the frames, when src and dest are on same port.

I can even do this.
VE1028# ping 10.1.28.4 & ping 10.1.28.101
PING 10.1.28.4 (10.1.28.4) 56(84) bytes of data.
[1] 3472
PING 10.1.28.101 (10.1.28.101) 56(84) bytes of data.
64 bytes from 10.1.28.4: icmp_seq=1 ttl=64 time=0.284 ms
64 bytes from 10.1.28.4: icmp_seq=2 ttl=64 time=0.207 ms
64 bytes from 10.1.28.4: icmp_seq=3 ttl=64 time=0.130 ms
64 bytes from 10.1.28.4: icmp_seq=4 ttl=64 time=0.175 ms
>From 10.1.28.12 icmp_seq=1 Destination Host Unreachable
>From 10.1.28.12 icmp_seq=2 Destination Host Unreachable
>From 10.1.28.12 icmp_seq=3 Destination Host Unreachable
64 bytes from 10.1.28.4: icmp_seq=5 ttl=64 time=0.176 ms
64 bytes from 10.1.28.4: icmp_seq=6 ttl=64 time=0.128 ms
64 bytes from 10.1.28.4: icmp_seq=7 ttl=64 time=0.173 ms
>From 10.1.28.12 icmp_seq=5 Destination Host Unreachable
>From 10.1.28.12 icmp_seq=6 Destination Host Unreachable
>From 10.1.28.12 icmp_seq=7 Destination Host Unreachable

Why does the bridge forward some frames
and block others to the same mac addr?

If I remove one physical interface from the bond, I have no problems
VE0# ifenslave -d bond0 eth1

continued output from VE1028...
64 bytes from 10.1.28.101: icmp_seq=8 ttl=64 time=1.04 ms
64 bytes from 10.1.28.4: icmp_seq=8 ttl=64 time=0.160 ms
64 bytes from 10.1.28.101: icmp_seq=9 ttl=64 time=1.22 ms
64 bytes from 10.1.28.4: icmp_seq=9 ttl=64 time=0.215 ms

Regards,

Subject: Re: Problem with bonding, vlan, bridge, veth
Posted by kfh on Wed, 15 Nov 2006 10:35:53 GMT
View Forum Message <> Reply to Message

> Hi list,
Hi list, will reply myelf :-)

> I have a bonding/vlan/bridge/veth problem.
> Sometimes a bridge think a veth device move to another port.
> If I remove a physical interface from bond, the bridge behaves normally.
>
> Kernel 2.6.16 + openvz test020
> VE0 Ubuntu dapper/6.06LTS, IP 172.31.1.26 on VLAN 254
> VE1028 Debian stable/sarge/3.1, IP 10.1.28.12 on VLAN 28
>
> I have a server (vs5, VE0) using eth0 and eth1 in a bonding interface
> bond0. bond0 is on tagged vlan.
> I create a vlan device vlan254 on vlan 254. This is VE0 IP.
> For each VE (XX) I do
> create a vlan device vlanXX on vlan XX.
> create a bridge bvXX and add vlanXX to it.
> create a VE (VE10XX) using veth.
> VETH="ve10XX.0,aa:00:04:56:YY:ZZ,eth0,aa:00:04:57:YY:ZZ"
> add ve10XX.0 to the bridge.
> YY and ZZ are calculated from VEID number (VLAN + 1000)
>
> eth0 eth1
> \ /
> bond0
> / \ veth
> vlan254 vlanXX ve10XX.0 -- eth0 (ve10XX)
> VE0 \ /
> bvXX (bridge)
>

The drawing above is correct, but the part not drawed
is the important one.

eth0 and eth1 are each connected to a switch.
These are connected by trunk ports 1 and 2.
The bond interface (eth0 + eth1) is in active/backup mode.

When I ping 10.1.28.101 in vlan28 from ve1028 (10.1.28.12),
it sends the following arp request:
aa:00:04:57:04:04 > ff:ff:ff:ff:ff:ff arp who-has 10.1.28.101 tell 10.1.28.12

The request will go from eth0 (VE1028) to ve1028.0 -> bv28 -> vlan28 ->
bond0 -> eth0 -> SW1port16 -> SW1 ALL ports but 16 -> including SW2port1/2 ->
SW2 ALL ports but 1/2 -> including target and eth1 -> bond0 -> vlan28 ->
bv28 -> ve1028.0 -> eth0

The target 10.28.1.101, receives the request through SW2 port 6.
The switches/bridges gets updated as follows:
 bv28 know aa:00:04:57:04:04 is at port 2 (ve1028.0)
 SW1 know aa:00:04:57:04:04 is at port 16

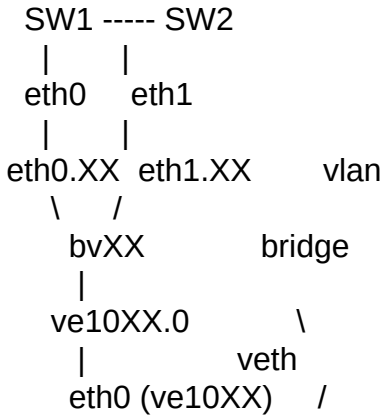
SW2 know aa:00:04:57:04:04 is at port 1/2
bv28 know aa:00:04:57:04:04 is at port 1 (vlan28)
Note bv28 gets updated twice.

The target replies:
00:03:fa:0f:a3:a7 > aa:00:04:57:04:04 arp reply 10.1.28.101 is-at ...:0f:a3:a7

The arp reply will go from SW2port6 -> SW2port1/2 -> SW1port1/2 ->
SW1port16 -> eth0 -> bond0 -> vlan28 -> bv28 -> NULL
As bv28 received the arp request from "aa:00:04:57:04:04" on port 1 (vlan28)
it will not forward the arp reply to port 2 (ve1028.0), therefore eth0 in
VE1028 never receives the arp reply... No communication.

So the problem is bridging over bonding.
The backup interface receives broadcast frames and forwards them to the bridge
which updates its mac table.

I will test the following.



I just have to make sure to use spanning tree.
The linux box should be in blocking mode.

Comments?

Regards,
Kristian.

Subject: Re: Problem with bonding, vlan, bridge, veth
Posted by dev on Wed, 15 Nov 2006 11:21:06 GMT
View Forum Message <> Reply to Message

Kristian,

thanks for sharing this info.

However, since it looks like your problem is related to bonding and bridges
(not OpenVZ itself) I think you would be able to get quicker/better reply from
netdev@vger>@kernel.org mailing list. Please, keep this mail list on CC.

Thanks,
Kirill

>
>>Hi list,
>
> Hi list, will reply myelf :-)
>
>
>>I have a bonding/vlan/bridge/veth problem.
>>Sometimes a bridge think a veth device move to another port.
>>If I remove a physical interface from bond, the bridge behaves normally.
>>
>>Kernel 2.6.16 + openvz test020
>>VE0 Ubuntu dapper/6.06LTS, IP 172.31.1.26 on VLAN 254
>>VE1028 Debian stable/sarge/3.1, IP 10.1.28.12 on VLAN 28
>>
>>I have a server (vs5, VE0) using eth0 and eth1 in a bonding interface
>>bond0. bond0 is on tagged vlan.
>>I create a vlan device vlan254 on vlan 254. This is VE0 IP.
>>For each VE (XX) I do
>> create a vlan device vlanXX on vlan XX.
>> create a bridge bvXX and add vlanXX to it.
>> create a VE (VE10XX) using veth.
>> VETH="ve10XX.0,aa:00:04:56:YY:ZZ,eth0,aa:00:04:57:YY:ZZ"
>> add ve10XX.0 to the bridge.
>> YY and ZZ are calculated from VEID number (VLAN + 1000)
>>
>>   eth0   eth1
>>   \   /
>>   bond0
>>   /   \   veth
>> vlan254   vlanXX   ve10XX.0 -- eth0 (ve10XX)
>>   VE0   \   /
>>   bvXX (bridge)
>>
>
>
> The drawing above is correct, but the part not drawed
> is the important one.
>
> eth0 and eth1 are each connected to a switch.
> These are connected by trunk ports 1 and 2.

> The bond interface (eth0 + eth1) is in active/backup mode.
>
> When I ping 10.1.28.101 in vlan28 from ve1028 (10.1.28.12),
> it sends the following arp request:
> aa:00:04:57:04:04 > ff:ff:ff:ff:ff:ff arp who-has 10.1.28.101 tell 10.1.28.12
>
> The request will go from eth0 (VE1028) to ve1028.0 -> bv28 -> vlan28 ->
> bond0 -> eth0 -> SW1port16 -> SW1 ALL ports but 16 -> including SW2port1/2 ->
> SW2 ALL ports but 1/2 -> including target and eth1 -> bond0 -> vlan28 ->
> bv28 -> ve1028.0 -> eth0
>
> The target 10.28.1.101, receives the request through SW2 port 6.
> The switches/bridges gets updated as follows:
> bv28 know aa:00:04:57:04:04 is at port 2 (ve1028.0)
> SW1 know aa:00:04:57:04:04 is at port 16
> SW2 know aa:00:04:57:04:04 is at port 1/2
> bv28 know aa:00:04:57:04:04 is at port 1 (vlan28)
> Note bv28 gets updated twice.
>
> The target replies:
> 00:03:fa:0f:a3:a7 > aa:00:04:57:04:04 arp reply 10.1.28.101 is-at ...:0f:a3:a7
>
> The arp reply will go from SW2port6 -> SW2port1/2 -> SW1port1/2 ->
> SW1port16 -> eth0 -> bond0 -> vlan28 -> bv28 -> NULL
> As bv28 received the arp request from "aa:00:04:57:04:04" on port 1 (vlan28)
> it will not forward the arp reply to port 2 (ve1028.0), therefore eth0 in
> VE1028 never receives the arp reply... No communication.
>
> So the problem is bridging over bonding.
> The backup interface receives broadcast frames and forwards them to the bridge
> which updates its mac table.
>
> I will test the following.
>
>
>   SW1 ----- SW2
> | |
>   eth0   eth1
> | |
>   eth0.XX eth1.XX   vlan
>   \   /
>   bvXX   bridge
> |
>   ve10XX.0   \
> | veth
>   eth0 (ve10XX)   /
>
> I just have to make sure to use spanning tree.

> The linux box should be in blocking mode.
>
> Comments?
>
> Regards,
> Kristian.
>

Subject: Re: Problem with bonding, vlan, bridge, veth
Posted by kfh on Thu, 23 Nov 2006 11:43:13 GMT
View Forum Message <> Reply to Message

On Wednesday den 15. November 2006 12:28, Kirill Korotaev wrote:
> Kristian,
>
> thanks for sharing this info.
> However, since it looks like your problem is related to bonding and bridges
> (not OpenVZ itself) I think you would be able to get quicker/better reply
> from netdev@vger>@kernel.org mailing list. Please, keep this mail list on
> CC.

I found the solution.
A patch added to git 20060304 has the following description:

- The current bonding driver receives duplicate packets when broadcast/
- multicast packets are sent by other devices or packets are flooded by the
- switch. In this patch, new flags are added in priv_flags of net_device
- structure to let the bonding driver discard duplicate packets in
- dev.c:skb_bond().

 http://www.kernel.org/git/?p=linux/kernel/git/torvalds/linux
-2.6.git;a=commit;h=8f903c708fcc2b579ebf16542bf6109bad593a1d

The "sad" part is the patch was the first applied to bonding after
the 2.6.16 release.

Regards,
Kristian.

> Thanks,
> Kirill
>

> >>Hi list,
> >
> > Hi list, will reply myelf :-)

> >
> >>I have a bonding/vlan/bridge/veth problem.
> >>Sometimes a bridge think a veth device move to another port.
> >>If I remove a physical interface from bond, the bridge behaves normally.
> >>
> >>Kernel 2.6.16 + openvz test020
> >>VE0 Ubuntu dapper/6.06LTS, IP 172.31.1.26 on VLAN 254
> >>VE1028 Debian stable/sarge/3.1, IP 10.1.28.12 on VLAN 28
> >>
> >>I have a server (vs5, VE0) using eth0 and eth1 in a bonding interface
> >>bond0. bond0 is on tagged vlan.
> >>I create a vlan device vlan254 on vlan 254. This is VE0 IP.
> >>For each VE (XX) I do
> >> create a vlan device vlanXX on vlan XX.
> >> create a bridge bvXX and add vlanXX to it.
> >> create a VE (VE10XX) using veth.
> >> VETH="ve10XX.0,aa:00:04:56:YY:ZZ,eth0,aa:00:04:57:YY:ZZ"
> >> add ve10XX.0 to the bridge.
> >> YY and ZZ are calculated from VEID number (VLAN + 1000)
> >>
> >>   eth0   eth1
> >>   \   /
> >>   bond0
> >>   /   \   veth
> >> vlan254   vlanXX   ve10XX.0 -- eth0 (ve10XX)
> >>   VE0   \   /
> >>   bvXX (bridge)
> >
> > The drawing above is correct, but the part not drawed
> > is the important one.
> >
> > eth0 and eth1 are each connected to a switch.
> > These are connected by trunk ports 1 and 2.
> > The bond interface (eth0 + eth1) is in active/backup mode.
> >
> > When I ping 10.1.28.101 in vlan28 from ve1028 (10.1.28.12),
> > it sends the following arp request:
> > aa:00:04:57:04:04 > ff:ff:ff:ff:ff:ff arp who-has 10.1.28.101 tell
> > 10.1.28.12
> >
> > The request will go from eth0 (VE1028) to ve1028.0 -> bv28 -> vlan28 ->
> > bond0 -> eth0 -> SW1port16 -> SW1 ALL ports but 16 -> including
> > SW2port1/2 -> SW2 ALL ports but 1/2 -> including target and eth1 -> bond0
> > -> vlan28 -> bv28 -> ve1028.0 -> eth0
> >
> > The target 10.28.1.101, receives the request through SW2 port 6.
> > The switches/bridges gets updated as follows:
> > bv28 know aa:00:04:57:04:04 is at port 2 (ve1028.0)

> > SW1 know aa:00:04:57:04:04 is at port 16
> > SW2 know aa:00:04:57:04:04 is at port 1/2
> > bv28 know aa:00:04:57:04:04 is at port 1 (vlan28)
> > Note bv28 gets updated twice.
> >
> > The target replies:
> > 00:03:fa:0f:a3:a7 > aa:00:04:57:04:04 arp reply 10.1.28.101 is-at
> > ...:0f:a3:a7
> >
> > The arp reply will go from SW2port6 -> SW2port1/2 -> SW1port1/2 ->
> > SW1port16 -> eth0 -> bond0 -> vlan28 -> bv28 -> NULL
> > As bv28 received the arp request from "aa:00:04:57:04:04" on port 1
> > (vlan28) it will not forward the arp reply to port 2 (ve1028.0),
> > therefore eth0 in VE1028 never receives the arp reply... No
> > communication.
> >
> > So the problem is bridging over bonding.
> > The backup interface receives broadcast frames and forwards them to the
> > bridge which updates its mac table.
> >
> > I will test the following.
> >
> >
> > SW1 ----- SW2
> >
> > eth0 eth1
> >
> > eth0.XX eth1.XX vlan
> > \ /
> > bvXX bridge
> >
> > ve10XX.0 \
> >
> > | veth
> >
> > eth0 (ve10XX) /
> >
> > I just have to make sure to use spanning tree.
> > The linux box should be in blocking mode.
> >
> > Comments?
> >
> > Regards,
> > Kristian.
> >

Subject: Re: Problem with bonding, vlan, bridge, veth
Posted by kir on Thu, 23 Nov 2006 13:12:25 GMT
View Forum Message <> Reply to Message

Have you tried OpenVZ 2.6.18-based kernel yet? Perhaps it has that patch
already...

Kristian F. Høgh wrote:
> On Wednesday den 15. November 2006 12:28, Kirill Korotaev wrote:
>
>> Kristian,
>>
>> thanks for sharing this info.
>> However, since it looks like your problem is related to bonding and bridges
>> (not OpenVZ itself) I think you would be able to get quicker/better reply
>> from netdev@vger>@kernel.org mailing list. Please, keep this mail list on
>> CC.
>>
>
> I found the solution.
> A patch added to git 20060304 has the following description:
>
> - The current bonding driver receives duplicate packets when broadcast/
> - multicast packets are sent by other devices or packets are flooded by the
> - switch. In this patch, new flags are added in priv_flags of net_device
> - structure to let the bonding driver discard duplicate packets in
> - dev.c:skb_bond().
>
> http://www.kernel.org/git/?p=linux/kernel/git/torvalds/linux
-2.6.git;a=commit;h=8f903c708fcc2b579ebf16542bf6109bad593a1d
>
> The "sad" part is the patch was the first applied to bonding after
> the 2.6.16 release.
>
> Regards,
> Kristian.
>
>
>
>> Thanks,
>> Kirill
>>
>>
>>> On Friday den 10. November 2006 11:12, Kristian F. Høgh wrote:
>>>
>>>> Hi list,
>>>>
>>> Hi list, will reply myelf :-)
>>>

>>>
>>>> I have a bonding/vlan/bridge/veth problem.
>>>> Sometimes a bridge think a veth device move to another port.
>>>> If I remove a physical interface from bond, the bridge behaves normally.
>>>>
>>>> Kernel 2.6.16 + openvz test020
>>>> VE0 Ubuntu dapper/6.06LTS, IP 172.31.1.26 on VLAN 254
>>>> VE1028 Debian stable/sarge/3.1, IP 10.1.28.12 on VLAN 28
>>>>
>>>> I have a server (vs5, VE0) using eth0 and eth1 in a bonding interface
>>>> bond0. bond0 is on tagged vlan.
>>>> I create a vlan device vlan254 on vlan 254. This is VE0 IP.
>>>> For each VE (XX) I do
>>>> create a vlan device vlanXX on vlan XX.
>>>> create a bridge bvXX and add vlanXX to it.
>>>> create a VE (VE10XX) using veth.
>>>> VETH="ve10XX.0,aa:00:04:56:YY:ZZ,eth0,aa:00:04:57:YY:ZZ"
>>>> add ve10XX.0 to the bridge.
>>>> YY and ZZ are calculated from VEID number (VLAN + 1000)
>>>>
>>>>   eth0   eth1
>>>>   \   /
>>>>   bond0
>>>>   /   \   veth
>>>> vlan254   vlanXX   ve10XX.0 -- eth0 (ve10XX)
>>>>   VE0   \   /
>>>>   bvXX (bridge)
>>>>
>>> The drawing above is correct, but the part not drawed
>>> is the important one.
>>>
>>> eth0 and eth1 are each connected to a switch.
>>> These are connected by trunk ports 1 and 2.
>>> The bond interface (eth0 + eth1) is in active/backup mode.
>>>
>>> When I ping 10.1.28.101 in vlan28 from ve1028 (10.1.28.12),
>>> it sends the following arp request:
>>> aa:00:04:57:04:04 > ff:ff:ff:ff:ff:ff arp who-has 10.1.28.101 tell
>>> 10.1.28.12
>>>
>>> The request will go from eth0 (VE1028) to ve1028.0 -> bv28 -> vlan28 ->
>>> bond0 -> eth0 -> SW1port16 -> SW1 ALL ports but 16 -> including
>>> SW2port1/2 -> SW2 ALL ports but 1/2 -> including target and eth1 -> bond0
>>> -> vlan28 -> bv28 -> ve1028.0 -> eth0
>>>
>>> The target 10.28.1.101, receives the request through SW2 port 6.
>>> The switches/bridges gets updated as follows:
>>> bv28 know aa:00:04:57:04:04 is at port 2 (ve1028.0)

>>> SW1 know aa:00:04:57:04:04 is at port 16
>>> SW2 know aa:00:04:57:04:04 is at port 1/2
>>> bv28 know aa:00:04:57:04:04 is at port 1 (vlan28)
>>> Note bv28 gets updated twice.
>>>
>>> The target replies:
>>> 00:03:fa:0f:a3:a7 > aa:00:04:57:04:04 arp reply 10.1.28.101 is-at
>>> ...:0f:a3:a7
>>>
>>> The arp reply will go from SW2port6 -> SW2port1/2 -> SW1port1/2 ->
>>> SW1port16 -> eth0 -> bond0 -> vlan28 -> bv28 -> NULL
>>> As bv28 received the arp request from "aa:00:04:57:04:04" on port 1
>>> (vlan28) it will not forward the arp reply to port 2 (ve1028.0),
>>> therefore eth0 in VE1028 never receives the arp reply... No
>>> communication.
>>>
>>> So the problem is bridging over bonding.
>>> The backup interface receives broadcast frames and forwards them to the
>>> bridge which updates its mac table.
>>>
>>> I will test the following.
>>>
>>>
>>>   SW1 ----- SW2
>>>
>>>   eth0   eth1
>>>
>>>   eth0.XX eth1.XX   vlan
>>>   \   /
>>>   bvXX   bridge
>>>
>>>   ve10XX.0   \
>>>
>>> | veth
>>>
>>>   eth0 (ve10XX)   /
>>>
>>> I just have to make sure to use spanning tree.
>>> The linux box should be in blocking mode.
>>>
>>> Comments?
>>>
>>> Regards,
>>> Kristian.
>>>

Subject: Re: Problem with bonding, vlan, bridge, veth
Posted by kfh on Thu, 23 Nov 2006 14:03:42 GMT
View Forum Message <> Reply to Message

On Thursday den 23. November 2006 14:12, Kir Kolyshkin wrote:
> Have you tried OpenVZ 2.6.18-based kernel yet? Perhaps it has that patch
> already...

No. Upstream does have it from 2.6.17, so I guess it's there.
I have applied the patch myself to "my" 2.6.16-openvz.

/Kristian.

<snip>