Subject: Dedicating physical NIC ports to each VE without bridging, etc.
Posted by dranch on Fri, 17 Aug 2007 02:06:31 GMT
View Forum Message <> Reply to Message

Hello Everyone,

I'm looking to see if there is any way to physically dedicate
a physical NIC port to each VE. If I can do this, it should
remove all the various complexities of troubleshooting
bridging, etc. The main reason for this is because I'm worried
about all the issues I might see when my users are messing with various things in the VEs such as altering MTUs (smaller/jumbo), VLAN tags, etc.

Xen can do dedicated NIC mode this via a hacky way of
excluding the PCI ID's from the Dom0 kernel and allowing the DomU kernels to discover then on their own. But.. I don't want to
run Xen.

For a specific example, say I have the following.

A physical machine with four ethernet ports (eth0-3) and I want:

physical port	Mapped location
eth0	base machine for management and access to all other VEs for management only
eth1	VE 100 - eth0
eth2	VE 101 - eth0
eth3	VE 102 - eth0

According to the following URL posted on the OpenVZ site, page 68: This *is* possible:

http://download.openvz.org/doc/OpenVZ-Users-Guide.pdf

"Still, there is a possibility for a VPS to directly access
a physical network adapter (for example, eth1). In this
case the adapter becomes inaccessible to the Hardware
Node itself."

vzctl set 100 --netdev_add eth1 --save

When I do this, it's doing two things:

1. For VE#100, it's actually mapping this interface in
   VE#100 as eth1 when I really want it mapped as eth0.
   This could give be an eth0 in every VE. The manpage
   for vzctl says that all other options for netdev_add
   are optional but whenever I give the full syntax,
   it complains about the VE's MAC address being an
   incorrect legth. So I altered the resulting
   /etc/vz/conf/100.conf file as follows and
   that help map things to eth0 in VE#100 but it's not
   doing what I want (see item #2):

NETIF=" ifname=eth0,mac=00:18:51:7C:1C:A6,host_ifname=veth100.1,host _mac=00:18:51
:1F:F4:0B"

2. When I "ifconfig eth1 up" the interface in VE#0 (master
   OS) with no IP assigned, then bring eth0 up in VE#100
   with say "ifconfig eth0 1.1.0.1 netmask 255.255.255.0",
   I don't seem to be able to transmit any data (no blinking
   LEDs). If I run tcpdump in VE#0 on interface
   "veth100.1", I *DO* see the ARP requests for my pings,
   etc.

Item #2 above gives me the impression that I *MUST* configure
bridging and map this veth100.1 interface and the eth1 interface into a unique bridge group within VE#0.

Is there any way to avoid this?

--David

Subject: Re: Dedicating physical NIC ports to each VE without bridging, etc.
Posted by den on Fri, 17 Aug 2007 05:55:02 GMT
View Forum Message <> Reply to Message

Basically, Ethernet device moved inside a VE is a normal Ethernet device visible to a VE. The device should "just work" after the move, i.e. you should assign an address to it and "up" it.

This means that all configuration staff you introduce is not needed. The only thing you need is to say
   vzctl set NNN --netdev_add eth2 --save
and appropriately modify network startup scripts inside a VE in a normal way like with the plain Linux.

We do not see any problem if the device will have eth1-eth4 names inside a VE. We believe that this is a normal practice.

Unfortunately, renaming a device inside a VE to eth0 is a bad policy. We should store old device

name on interface moving or you'll have a device name conflict on VE stop. So, device renaming interfaces are forbidden inside a VE.

There are some ways to rename an interface inside a VE or before an actual interface moving and we can describe them if you really need them.

Regards,
Den

Subject: Re: Dedicating physical NIC ports to each VE without bridging, etc.
Posted by dranch on Fri, 17 Aug 2007 21:12:21 GMT
View Forum Message <> Reply to Message

Ok, so after rebooting the machine, I did the following:

VE#0:
```
/sbin/ifconfig eth1 0  #UPs the int and removes any IPv4 addr
/sbin/ifconfig eth2 0
/sbin/ifconfig eth3 0
/usr/sbin/vzctl set 100 --netdev_add eth1 --save
/usr/sbin/vzctl set 101 --netdev_add eth2 --save
/usr/sbin/vzctl set 102 --netdev_add eth3 --save

/usr/sbin/vzctl start 100
```

When I start VE#100, I now see VE#0's eth1 disappear and I can now "ifconfig eth1 up" in VE#100 as hoped! Woohoo... excellent news and it seems to work from initiate testing. This helps remove any MTU / VLAN tagging / etc. issues I might have seen if I had to deal with bridging virtual to physical interfaces.

Now, the issue of not supporting "eth0" in the VEs is unfortunate. Linux always enumerates from 0 so not having eth0 but say eth2 in a VE is both a change in behavior as well as just confusing to me and I'm sure to my developers as well. I'd love to see the ability to rename VE interfaces if possible.

--David
