WEBlog -- Wouter's Eclectic Blog/ en/ computer/ nbd/

root-on-nbd

Start grepping:
Pente
Travel
Parsing accept-* in PHP
GNoDbify
G.Mate Yopy
Interview on Radio Centraal
My Debian involvement
Wouter's blog

Root on NBD, now with partitions

A while ago, I wrote an initrd script and hook that would allow one to run a system with root on a Network Block Device (NBD), that I maintain both upstream and in Debian.

Since then, Laurent Vivier submitted a kernel patch that got merged, and which will allow one to partition an NBD device. I had a patch for my initrd script that would support running root off a partitioned NBD device, but did not yet have the time to test and debug it.

Until today, that is. Well... not really. I *made* the time, but I didn't really have it. For reference, the time is 07:25 right now, and yes, I did stay up all night. But, well, who cares, right?

Right.

I'd really love to get this into lenny still, but since we're in deep freeze... not keeping my breath. We'll see, I guess. The upload is going to unstable, anyway—after this one final test.

So how does it work? Unfortunately, no d-i support as of yet. I'd been working on that, but then it turned out that in order to support root-on-NBD, I'd have to somehow, somewhere, teach partman that NBD is this strange thing which can be formatted, but not partitioned. I was way far in my research (but nowhere in code, yet) on that when Laurent sent in his partitioning patch, so it got postponed then, and I didn't get a chance to pick it up anymore in time.

But if you're willing to use debootstrap manually, it's really simple to set up:

```
# apt-get install nbd-client
# nbd-client server port /dev/nbd0
# fdisk /dev/nbd0
[... create partitions ...]
# mkfs -t ext3 /dev/nbd0p1
# mkdir /mnt/tmp
# mount /dev/nbd0p1 /mnt/tmp
# debootstrap --include=nbd-client sid /mnt/tmp
# echo KILLALL=false >> /mnt/tmp/etc/nbd-client
```

The "KILLALL" is necessary, since otherwise the nbd-client initscript will do nasty things with your root filesystem. Note that you still need nbd-client installed and run the initscript, even if it only uses nbd-client inside the initrd; because the initscript writes the nbd-client PID to `/lib/init/rw/sendsigs.omit.d`, which is *also* required if you don't want sendsigs to shoot the root filesystem from under your nose. This might see some changes in the future.

Now boot with your kernel and initramfs, and boot it with the following arguments:

```
nbdroot=server,port root=/dev/nbd0p1
```

Obviously, in the above two examples, you should replace 'server' by the IP address or hostname of your server, and 'port' by the port number on which your nbd-server is running. It is of course also possible to use nbd1p2 or nbd0p4, should you want that.

And that's all there is to it.

Well, almost; due to the way the network configuration is written in initramfs, I had to choose between reinventing the wheel, or hardcoding the NIC name to 'eth0'. I chose the latter. What this means is that if you want to boot off an NBD blockdevice that is only available on eth1 or wlan0 or some such, you lose.

Other than that, go ahead and try. Should be on a DMNY (Debian Mirror Near You) RSN.

Update: fill in the link to git.kernel.org, and mention the KILLALL thing. Oops.

Add a comment

Last edited Wed Dec 24 14:24:45 2008