Subject: Re:  Re: lvm and openvz
Posted by kir on Thu, 29 Mar 2012 12:16:47 GMT

On 03/29/2012 01:09 PM, David Brown wrote:
> On 29/03/2012 11:04, Kir Kolyshkin wrote:
>> On 03/28/2012 11:49 AM, David Brown wrote:
>>> I /really/ wish the openvz developers would move beyond kernel
>>> 2.6.32 - kernal 2.6.33 introduced snapshot merging to LVM which
>>> would play wonderfully with this setup.
>> I'm not sure why people think that RHEL6 kernel is pure 2.6.32. It is
>>   definitely not!
>>
>> For snapshot merging, I am not an expert here but googling for 'rhel6
>>   lvm snapshot merging' gave me this:
>>
>>  http://www.linuxtopia.org/online_books/rhel6/rhel_6_lvm_admi
n/rhel_6_lvm_snapshot_merge.html
>>
>>
>>
>> and this:
>>
>> https://access.redhat.com/knowledge/solutions/58510
>>
>> Both articles suggest RHEL6 kernel supports LVM snapshot merging, and
>> so should OpenVZ RHEL6-based kernel.
>>
>> PS If you are using non-rhel6 openvz kernel, it's definitely time to
>>   switch, and lots of reasons to do that besides LVM snapshot merging.
>> Notable things are vswap, ploop, stability...
>>
>> Kir.
> I am using a non-rhel6 openvz kernel because I don't use RHEL - I use
> Debian on my servers.  Are you suggesting that I should specifically use
> the RHEL6 openvz kernel even though I use Debian?  That's something I
> haven't thought of trying, but if it is the recommendation of the OpenVZ
> developers, then I will give it a shot.

Yes please. This is the official recommendation, and many people already
do this.

We choose RHEL6 kernel as a base not because we are Red Hat fans or we
plan to support
RHEL distribution only. The reason is RHEL6 is truly a good,
well-maintained, stable kernel with
lots of Red Hat developer and QA resources invested into it. That is why
we use it as a base,

but that doesn't mean we only have RHEL in mind.

Now to the practical point: we have modified post-install scripts in
kernel rpm to be compatible
with Debian as well, so all you need to do is to convert kernel rpm to
deb using alien. Some
info is provided at
 http://wiki.openvz.org/Install_kernel_from_RPM_on_Debian_6.0

> More generally, I would hope that one day OpenVZ will change over to
> following the current kernel (or perhaps the current long-term support
> kernels) - there has been a lot of development since 2.6.32, not all of
> which gets backported by Red Hat.

Same as we did before - we do have such plans, although not immediate.

> I'd expect that a lot of OpenVZ code
> can be merged with or replaced by the container support in later
> kernels.

That is right, we are actively pushing our stuff upstream, and then we
are rebasing our code to what is available in upstream, gradually
reducing the size of openvz patchset.

For example, if you will take a look at linux kernel git repo, you will
see more than about 150 patches from OpenVZ developers were merged this
year. Actually, here's a command to do that:

$ git log -E --author='@parallels.com|@openvz.org' --since=2012-01-01 |
grep -c ^commit

Most of the recent patches are CRUI and NFS virtualization.

>   I also think that if OpenVZ doesn't catch up, then people will
> migrate to other solutions such as Linux VServer or LXC (I know I
> considered it for the last server I configured).

Linux-VServer is totally obsolete from my POV, because they are not
interested in pushing the stuff upstream. Of course they benefit from
the code that is appearing in mainline (and since a good proportion of
that code comes from OpenVZ developers, it is true to say that
Linux-VServer benefits from OpenVZ).

As for the LXC, please do not forget that LXC is not something that is
opposed to OpenVZ, but rather something that is complementary. I mean,
having said that the good proportion of containers code in mainline
comes from OpenVZ, it might be true to say that we are probably the
biggest contributor to the LXC (kernel code).

> Of course, I fully appreciate that something like that takes a lot of
> effort, and that means time, money, people to do the work, testing,
> etc., etc.  But one can still hope!

Our current approach is to use RHEL kernels as a base, and push as much
stuff to upstream as we can. So far it's working.

---