

Saturday, March 22. 2008

## Less known Solaris Features: iSCSI - Part 4: Alternative backing stores

Albeit it's very easy to use iSCSI in conjunction with ZFS emulated volumes it doesn't have to be this way. You can use different kinds of backing store for iSCSI

File based iSCSI target

One way to provide for storage for an iSCSI target. You can use files for this task. Okay, we have to login as root on theoden:

```
# mkdir -p /var/iscsitargets
```

```
# iscsitadm modify admin -d /var/iscsitargets
```

At first we've created a directory to keep the files, then we tell the target daemon to use this for storing the target. After this we can create the target:

```
# iscsitadm create target --size 128m smalltarget
```

Now we can check for the iSCSI Target.

```
# iscsitadm list target -v smalltarget
```

```
Target: smalltarget
```

```
iSCSI Name: iqn.1986-03.com.sun:02:3898c6a7-1f43-e298-e189-83d10f88131d.smalltarget
```

```
Connections: 0
```

```
ACL list:
```

```
TPGT list:
```

```
LUN information:
```

```
LUN: 0
```

```
GUID: 0100001c42e9f21a00002a0047e45145
```

```
VID: SUN
```

```
PID: SOLARIS
```

```
Type: disk
```

```
Size: 128M
```

```
Status: offline
```

Now we switch to the server we use as an initiator. Let's scan for new devices on gandalf. As

```
we've activated the discovery of targets before, we've just have to scan for new devices.
```

```
# devfsadm -c iscsi -C
```

```
# format
```

Searching for disks...done

AVAILABLE DISK SELECTIONS:

```
0. c0d0
```

```
/pci@0,0/pci-ide@1f,1/ide@0/cmdk@0,0
```

```
1. c1d0
```

```
/pci@0,0/pci-ide@1f,1/ide@1/cmdk@0,0
```

```
2. c2t0100001C42E9F21A00002A0047E39E34d0
```

```
/scsi_vhci/disk@g0100001c42e9f21a00002a0047e39e34
```

```
3. c2t0100001C42E9F21A00002A0047E45145d0
```

```
/scsi_vhci/disk@g0100001c42e9f21a00002a0047e45145
```

```
Specify disk (enter its number):
```

```
Specify disk (enter its number): ^C
```

```
# Et voila, an additional LUN is available on our initiator.
```

Thin-provisioned target backing store

One nice thing about ZFS is it's ability to provide thin provisioned emulated zfs volumes (zvol). You can configure a volume of a larger size than the physical storage you have available. This is useful, when you want to have an volume in it's final size (because resizing would be a pain in the a...) but don't want do spend the money for the disks because you know that much less storage would be needed at first.

It's really easy to create such a kind of a zvol:

```
# zfs create -s -V 2g testpool/bigvolume
```

That's all. The difference is the small -s. It tells ZFS to create an sparse (aka thin) provisioned volume.

Well, I won't enable iSCSI for this by shareiscsi=on itself. I will configure this manually. As normal volumes zvols are available within the /dev tree of your filesystem:

```
# ls -l /dev/zvol/dsk/testpool
```

```
total 4
```

```
lrwxrwxrwx 1 root root 35 Mar 22 02:09 bigvolume -> ../../../../devices/pseudo/zfs@0:2c
```

```
lrwxrwxrwx 1 root root 35 Mar 21 12:33 zfsvolume -> ../../../../devices/pseudo/zfs@0:1c
```

Okay, we can use this

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

devices as a backing store for an iSCSI target as well. We've created a zvol bigvolume within the zpool testpool. Thus the device is /dev/zvol/dsk/testpool/bigvolume:# iscsitadm create target -b /dev/zvol/dsk/testpool/bigvolume bigtargetOkay, i'm switching to my root shell on the initiator. Again we scan for devices:# devfsadm -c iscsi -C # format  
Searching for disks...done

### AVAILABLE DISK SELECTIONS:

0. c0d0  
/pci@0,0/pci-ide@1f,1/ide@0/cmdk@0,0
1. c1d0  
/pci@0,0/pci-ide@1f,1/ide@1/cmdk@0,0
2. c2t0100001C42E9F21A00002A0047E39E34d0  
/scsi\_vhci/disk@g0100001c42e9f21a00002a0047e39e34
3. c2t0100001C42E9F21A00002A0047E45DB2d0  
/scsi\_vhci/disk@g0100001c42e9f21a00002a0047e45db2

Let's create an zpool:# zpool create zfsviaiscsi\_thin c2t0100001C42E9F21A00002A0047E45DB2d0  
# zpool list

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
------	------	------	-------	-----	--------	---------

zfsviaiscsi	187M	112K	187M	0%	ONLINE	-
-------------	------	------	------	----	--------	---

zfsviaiscsi_thin	1.98G	374K	1.98G	0%	ONLINE	-Do you remember, that we used four 128 MB files as the devices for our zpool on our target. Well, you have an 1.98G filesystem running on this files. You can add more storage to the zpool on the target and you have nothing to do on the initiator. Not a real kicker for ZFS, but imagine the same for other filesystem that can't be grown so easy like a zpool.
------------------	-------	------	-------	----	--------	--

Posted by Joerg Moellenkamp in English, Solaris at 11:37