Sunday, March 5. 2017

Slapped by metaslabs

I just want to share something that I've learned a few days ago by doing this. Let's assume you have a ZFS pool. 1 TB in size. You want to add some storage. By accident you grab a slice 0 that is 128MB large instead of the whole LUN. The obvious question is how do you get rid of it. You may get to the idea that you replace the 128 MB LUN with a 1 TB LUN. We do this replacement all the time to increase of rpools. However: For the given situation this is an exceptionally bad idea.

I demonstrate this issue with an example:

```
root@solaris:~# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1t0d0
          /pci@0,0/pci8086,2829@d/disk@0,0
       1. c1t2d0
          /pci@0,0/pci8086,2829@d/disk@2,0
       2. c1t3d0
          /pci@0,0/pci8086,2829@d/disk@3,0
       3. c1t4d0
          /pci@0,0/pci8086,2829@d/disk@4,0
Specify disk (enter its number): ^C
```

Okay, the pool was created like this:

```
root@solaris:~# zpool create datapool c1t3d0
```

Yeah, a single device pool. I know this isn't a good idea, but the customer configured it this way.

Before going forward you have to keep in mind that there is something called metaslabs. They are part of the way ZFS is organizing the storage is uses. They are really important to the code that is tracking the free parts of your disk. There are several good articles on this topic out there. At the moment keep in mind that Solaris aims to have 200 metaslabs on it.

Now let's look at the metaslabs of the first vdev (vdev 0) with zdb -m datapool 0:

```
vdev        0  ms_array        27
metaslabs  127  offset                spacemap         free
--------------  ----------------------  ---------------  -------------
       metaslab     0  offset              0  spacemap     30  free     128M
```

The metaslabs are sized 128 MB and you have 127 of them.

```
root@solaris:~# zdb -m datapool 0 | grep "metaslab" | wc -l
    128
```

Then the customer wanted more capacity and configured a LUN on their storage. However by accident he didn't add the 1 TB LUN he wanted to add but a 128 MB slice on it. Essentially what happened was something like that:

```
root@solaris:~# zpool add datapool c1t2d0
```

When you look for the metaslabs, you will see the metaslabs of the second vdev (vdev 1) are now 1 Megabyte in size.

```
root@solaris:~# zdb -m datapool 1 | grep "metaslab" | head -n 3
```

```
metaslabs  115  offset              spacemap         free
metaslab     0  offset                0  spacemap     33  free    1013K
metaslab     1  offset         0x100000  spacemap      0  free       1M
```

That's expected because this is exactly the behaviour we see see written down in the respective source code.

So, how do you get rid of this 128 MB device. You shouldn’t follow your first guess. Don’t replace the device. Just don’t do it. Why? Well … in principle it works. You can get rid of it. But it has consequences. Let’s just do it for demonstration.

```
root@solaris:~# zpool replace datapool c1t2d0 c1t4d0
```

The problem is: ZFS keeps the metaslab size on the vdev, it just creates more of them. I was aware of that for resizing a LUN, but hadn’t in mind that for replace it’s the same.

```
root@solaris:~# zdb -m datapool 1 | grep "metaslab" | head -n 3
      metaslabs16371  offset              spacemap         free
      metaslab     0  offset                0  spacemap     33  free    1006K
      metaslab     1  offset         0x100000  spacemap      0  free       1M
```

Well … and now extrapolate it to a 1 TB device that replaces a 128 MB device. You will have 1.000.000 metaslabs. This has a lot consequences on behaviour of this mechanism. It was written mit 200 metaslabs in mind per vdev ... not a million. The problems range from memory consumption over loading and unloading of metaslabs up to locking. Let's say it this way. You will have some performance problems when writing to it.

So ... do not replace the devices in this siutation. Just don’t do it. That said: This isn't a problem in normal operation. You increase from 600 GB to 1.2 TB or from 2TB to 4TB and then to 8 TB. So you end up with perhaps 800 Metaslabs. Not a problem. But not with a million.

And before you ask: Adding the large disk as a mirror and split the old one away doesn’t work as well.

```
root@solaris:~# zpool create datapool c1t3d0
root@solaris:~# zpool set autoexpand=on datapool
root@solaris:~# zpool add datapool c1t2d0
root@solaris:~# zpool attach datapool c1t2d0 c1t4d0
root@solaris:~# zpool detach datapool c1t2d0
root@solaris:~# zdb -m datapool 1 | grep "metaslab" | head -n 3
      metaslabs16371  offset              spacemap         free
      metaslab     0  offset                0  spacemap     33  free    1006K
      metaslab     1  offset         0x100000  spacemap      0  free       1M
root@solaris:~#
```

You still end up with 16371 metaslabs. Or a million when using a 1 TB disk.

So whats the solution for the accidentially added device: From my perspective ? Just leave it there at the moment. Don’t do anything. Recreate the pool with zfs send/receive. But do not simply replace the device.

Posted by Joerg Moellenkamp in English, Solaris at 16:58