Wednesday, January 8. 2014

Beware, biting bsize

There are a lot of suggestions to increase the rsize and the wsize parameter for NFS to get better performance when doing large transfers. The idea is that NFS is transmitting larger chunks at once and thus improving the performance. However there is a step that you have to do before, when you want to increase both parameters.

The situation
Okay, let's start with a small size. I prepared a NFS server VM with a single share on 10.10.10.1. So i'm mounting the share with 8k rsize/wsize just for a start:

```
root@nfsclient:/export/home/jmoekamp# mount -o rsize=8192,wsize=8192,vers=3
10.10.10.1:/export/home/jmoekamp/justashare /export/home/jmoekamp/justamountedshare
```

Now i'm executing a small test via dd in the mounted directory.

```
jmoekamp@nfsclient:~$ dd if=/dev/zero of=justamountedshare/test bs=1024k count=4
4+0 records in
4+0 records out
```

Okay. I'm using a small dtrace script in order to find out, what has been used by the client to transport the data to the server by looking at the stuff hitting the server. It's an really extremely simple dtrace script:

```
#!/usr/sbin/dtrace -s
#pragma D option quiet
#pragma D option switchrate=10hz

nfsv3:::op-write-start
{
 @count_w[args[2]->count] = count();
}
.
```

Okay, i should explain now, that I've started it before running the dd and stopped it afterwards:

```
root@nfsserver:~# ./writeblocksizecount.d
^C
    8192             512
```

Okay, the client transmitted 512 8k chunks to the server. As expected. 512 times 4k is 4 Megabyte. 4 times 1024k is 4 Megabyte. Now we repeat the same test with 32k chunks:

```
root@nfsclient:/export/home/jmoekamp# mount -o rsize=32768,wsize=32768,vers=3
10.10.10.1:/export/home/jmoekamp/justashare /export/home/jmoekamp/justamountedshare
jmoekamp@nfsclient:~$ dd if=/dev/zero of=justamountedshare/test bs=1024k count=4
4+0 records in
4+0 records out
root@nfsserver:~# ./writeblocksizecount.d
^C
   32768             128
```

128 chunks with 32k. Again ... as expected. Okay ... let's test bigger chunks ... 1 megabyte.

```
root@nfsclient:/export/home/jmoekamp# mount -o rsize=1048576,wsize=1048576,vers=3
10.10.10.1:/export/home/jmoekamp/justashare /export/home/jmoekamp/justamountedshare
jmoekamp@nfsclient:~$ dd if=/dev/zero of=justamountedshare/test bs=1024k count=4
4+0 records in
4+0 records out
```

```
root@nfsserver:~# ./writeblocksizecount.d
^C
   32768              128
```

WTF? Still 128 chunks with 32k each? The setting had no impact.

Solution
So ... what's the issue? Just because you specified a rsize and a wsize doesn't automatically imply that those chunk size are used. There are certain defaults in the system that limit the size of the chunks transported by NFS as well.

The parameters are called nfs:nfs3_bsize and nfs:nfs3_max_transfer_size. The bsize defaults to 32768. In Solaris 10 max_transfer_size defaults to 32768, it has changed to 1M in Solaris 11. For NFS4 there are separate settings with nfs:nfs4_bsize and nfs:nfs4_max_transfer_size

So by the default the maximum size of a chunk of data transported by NFS is 32768 ... no matter what you use as wsize and rsize. You have to change this defaults. nfs:nfs3_bsize has to be equal or larger than the maximum rsize or wsize you specify on the client. nfs:nfs3_max_transfer_size has to be equal or larger than nfs:nfs3_bsize.

One of the reason for this is the amount of data you have to allocate for doing communication. 32768 is compromise of minimising memory allocation and maximising performance for sequential loads. 8k would be better for memory allocation, 1M is better for sequential performance.

You can change the defaults. Easiest way is by editing /etc/system and rebooting:

```
root@nfsclient:/# echo "set nfs:nfs3_bsize=1048576" >> /etc/system
root@nfsclient:/# echo "set nfs:nfs3_max_transfer_size=1048576" >> /etc/system
root@nfsclient:/# reboot
```

Okay, let's do the test again:

```
root@nfsclient:~#  mount -o rsize=1048576,wsize=1048576,vers=3 10.10.10.1:/export/home/jmoekamp/justashare
/export/home/jmoekamp/justamountedshare
jmoekamp@nfsclient:~$ dd if=/dev/zero of=justamountedshare/test bs=1024k count=4
4+0 records in
4+0 records out
```

The dtrace script shows a different output now:

```
root@nfsserver:~# ./writeblocksizecount.d
^C
  1048576                4
```

As expect you see 4 chunks with a size 1048676 bytes. As configured by the rsize and wsize.

By the way: While a rsize/wsize of 1M may give you best single user sequential read/write performance, my experience so far suggests that 128k are a much better choice when several user are using the the mounted directory in parallel because the small size of the chunks of data to transmit allow the system to share the network link much better between all the requests initiated by the users.

Posted by Joerg Moellenkamp in English, Solaris at 23:39