

Monday, March 24, 2008

## Less Known Solaris features: SamFS - Part 7: Working with SamFS

Now we've configured a file system and we set up the archiver for it. Now let's use it.

Looking up SamFS specific metadataAt first let's create a test file.

```
# mkfile 10m /samfs1/testfile3We now look at the metadata of this file. There is a special version of ls that is capable to read the additional information. This version ls is called sls. So let's check for our test file. [root@elrond:/samfs1]$ sls -D testfile3
```

```
testfile3:
mode: -rw-----T links: 1 owner: root group: root
length: 1048576 admin id: 0 inode: 4640.1
access: Mar 23 19:28 modification: Mar 23 19:28
changed: Mar 23 19:28 attributes: Mar 23 19:28
creation: Mar 23 19:28 residence: Mar 23 19:28
```

There is nothing new. Okay, let's leave the computer alone, drink a coffee or two, and now we check again: bash-3.00#

```
sls -D testfile3
```

```
testfile3:
mode: -rw-----T links: 1 owner: root group: root
length: 1048576 admin id: 0 inode: 4640.1
archdone;
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
access: Mar 23 19:28 modification: Mar 23 19:28
changed: Mar 23 19:28 attributes: Mar 23 19:28
creation: Mar 23 19:28 residence: Mar 23 19:28
```

I assume you've already noticed the three additional lines. The archiver did it's job: archdone;

```
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
```

The first line says, that all outstanding archiving for the file is done. The two next lines tells you where the copies are located, when they were archived and tells you about some special flags. The 1.1 means first file in the archive file, starting at the 513th bit of the archive file (the header of tar is 512 byte long, thus the 513th bit is the first usable byte, thus the 1)

Manually forcing the releaseNormally a file just get released, when the high watermark is reached or you configured the archiving this way. But you can force the release on the command line.

After this step, the file isn't in the cache any longer. When we look in the metadata, we will the a new information. The file is in the offline state:

```
bash-3.00# sls -D testfile3
```

```
testfile3:
mode: -rw-----T links: 1 owner: root group: root
length: 1048576 admin id: 0 inode: 4640.1
offline; archdone;
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
access: Mar 23 19:28 modification: Mar 23 19:28
changed: Mar 23 19:28 attributes: Mar 23 19:28
creation: Mar 23 19:28 residence: Mar 24 01:28
```

When we access it again, the file get's staged back to the cache again: bash-3.00# cat testfile3

```
bash-3.00# sls -D testfile3
```

```
testfile3:
mode: -rw-----T links: 1 owner: root group: root
length: 1048576 admin id: 0 inode: 4640.1
archdone;
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
access: Mar 24 01:35 modification: Mar 23 19:28
changed: Mar 23 19:28 attributes: Mar 23 19:28
```

creation: Mar 23 19:28 residence: Mar 24 01:35The offline flag has gone away.

Manually forcing the staging of a fileOkay, but you can force the staging as well, let's assume you've released a file.

```
bash-3.00# release testfile3
```

```
bash-3.00# sls -D testfile3
```

```
testfile3:
```

```
mode: -rw-----T links: 1 owner: root group: root
```

```
length: 1048576 admin id: 0 inode: 4640.1
```

```
offline; archdone;
```

```
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
```

```
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
```

```
access: Mar 24 01:35 modification: Mar 23 19:28
```

```
changed: Mar 23 19:28 attributes: Mar 23 19:28
```

creation: Mar 23 19:28 residence: Mar 24 01:37A colleague comes into your office, and tells you that he wants to use a large file with simulation data tomorrow. It would be nice, if he don't have to wait for the automatic staging. We can force SamFS to get the file back to the cache.

```
bash-3.00# stage testfile3
```

```
testfile3:
```

```
mode: -rw-----T links: 1 owner: root group: root
```

```
length: 1048576 admin id: 0 inode: 4640.1
```

```
archdone;
```

```
copy 1: ----- Mar 23 19:39 1.1 dk disk01 f1
```

```
copy 2: ----- Mar 23 19:58 1.1 dk disk03 f1
```

```
access: Mar 24 01:35 modification: Mar 23 19:28
```

```
changed: Mar 23 19:28 attributes: Mar 23 19:28
```

```
creation: Mar 23 19:28 residence: Mar 24 01:37Voila, it's in the cache again.
```

Posted by Joerg Moellenkamp in Solaris at 21:03