

Thursday, April 24, 2008

Less Known Solaris features: fssnap

An ever reoccurring problem while doing backups is the problem, that you have to keep the state of the backup consistent. For example: You use the cyrus imapd for storing mails on a system. As it's a system used for years, the message store is still on an UFS file system. Okay, now there is a little problem: You want to make make backups, but it's a mail server.

With the amount of mail junkies in modern times, you can't take the system offline for an hour or so, just to make a backup. But you have to take it offline. Especially a mail server is a moving target, as cyrus imapd has indexes for its mailboxes, and index files for the mailboxes itself. Let's assume a backup takes 1 hour, and users delete mails in this time, create mailboxes and so on. It's possible that you backup your mailservier in an inconsistent state, das the mail directory of the user may represent the state one hour ago, but the mailboxes file represent the state of one minute ago.

fssnapUFS has a little known feature, that comes to help in such a situation. You can do a filesystem snapshot of the system. This is a non-changing point-in-time view to the filesystem, while you can still change the the original filesystem. fssnap is a rather old tool. We've introduced in the 1/01 update of Solaris 8.

There is a restriction with this snapshots: This snapshots are solely for the purpose of backups, thus they are not boot persistent. For boot persistent snapshots of a filesystem you will need the Sun Availability Suite.

A practical example. Let's assume that we have a directory called /mailserver. This directory is the mountpoint for a UFS filesystem on /dev/dsk/c1d1s1. At the moment it contains some files: # ls -ls

```
total 10
 2 -rw-----T 1 root  root    1024 Apr 23 01:41 testfile1
 2 -rw-----T 1 root  root    1024 Apr 23 01:41 testfile2
 2 -rw-----T 1 root  root    1024 Apr 23 01:41 testfile3
 2 -rw-----T 1 root  root    1024 Apr 23 01:41 testfile4
 2 -rw-----T 1 root  root    1024 Apr 23 01:41 testindex1
```

Now we want to make a backup. It's sensible to take the mail server offline for a few seconds to keep the files consistent. In this moment we make the filesystem snapshot. This is really easy: # fssnap -o bs=/tmp /mailserver /dev/fssnap/0

With this command we told fssnap to take an snapshot of the filesystem mounted at /mailserver. Furthermore we configured that the snapshot uses the /tmp for it's backing store. In the backing store the changes since the snapshot will be recorded. When fssnap is able to create the snapshot, it will return the name for the pseudo device containing the filesystem snapshot. In our case it's /dev/fssnap/0. Please remember it, we need it later.

When you look at the /tmp directory you will find an backing store file for this snapshot. It's called snapshot0 for the first snapshot on the system: # ls -l /tmp

```
total 910120
-rw-r--r-- 1 root  root    30 Apr 22 14:50 ogl_select305
```

```
-rw----- 1 root  root  465976320 Apr 23 01:47 snapshot0
```

Now we bring the mailserver online again, and after a few seconds we see changes to the filesystem again (okay, in my example i will do this manually): # mkfile 1k testfile5

```
# mkfile 1k testfile6
# mkfile 2k testindex1
```

```
# ls -l
total 16
-rw-----T 1 root  root    1024 Apr 23 01:41 testfile1
-rw-----T 1 root  root    1024 Apr 23 01:41 testfile2
-rw-----T 1 root  root    1024 Apr 23 01:41 testfile3
-rw-----T 1 root  root    1024 Apr 23 01:41 testfile4
-rw-----T 1 root  root    1024 Apr 23 01:53 testfile5
-rw-----T 1 root  root    1024 Apr 23 01:53 testfile6
-rw-----T 1 root  root   2048 Apr 23 01:53 testindex1
```

Now we want to make the backup itself. At first we have to mount the filesystem. Thus we create a mountpoint # mkdir /mailserver_forbackup

Now we mount the snapshot. You will recognize the pseudo device here again. The snapshot is read only thus you have to specify it at mount time: # mount -o ro /dev/fssnap/0 /mailserver_forbackup

Okay, when we look into the filesystem, we will see the state of the filesystem at the moment of the snapshot. testfile5, testfile6 and the bigger testindex1 are missing. # ls -l

total 10

-rw-----T 1 root root 1024 Apr 23 01:41 testfile1

-rw-----T 1 root root 1024 Apr 23 01:41 testfile2

-rw-----T 1 root root 1024 Apr 23 01:41 testfile3

-rw-----T 1 root root 1024 Apr 23 01:41 testfile4

-rw-----T 1 root root 1024 Apr 23 01:41 testindex1
Now we can do a backup without any problems and in a consistent manner:
tar cfv /backup_mailserver_20080424.tar *

a testfile1 1K

a testfile2 1K

a testfile3 1K

a testfile4 1K

a testindex1 1K
After this step we should clean-up. We unmount the snapshot and delete the snapshot with its backing store file:
umount /mailserver_forbackup

fssnap -d /mailserver

Deleted snapshot 0.

rm /tmp/snapshot0

Conclusion
With the fssnap command you have an easy way to do consistent backups on UFS filesystems. While it's not as powerful as the functions of the point-in-time copy functionality in the Availability Suite, it's a perfect match for its job.

Do you want to learn more?
fssnap(1M)

fssnap_ufs(1M)

Posted by Joerg Moellenkamp in English at 07:04