

Monday, August 18, 2008

## Less known Solaris Features: CacheFS - Part 5: On-demand consistency checking

Let's assume you share a filesystem with static content (for example the copy of a cdrom) or a filesystem that changes on a regular schedule (for example at midnight every day). So it would pose unnecessary load to network to check the consistency everytime again.

CacheFS knows a special mode of operation for such an situation. It's called on demand consistency checking. It does exactly what the name says: It checks only the consistency of files in the cache, when you tell the system to do it.

I will demonstrate this with an example:

With normal consistency checkLet's assume, we stay with the normal mode of operation from the example before. We create a file on the fileserver.

```
[root@theoden:/export/files]# date >> test_with_consistency_check
[root@theoden:/export/files]# cat test_with_consistency_check
Tue Aug 12 14:59:54 CEST 2008When we go to the NFS client and access the directory, this new file is visible
instantaneous. And when we access it, we see the content of the file.[root@gandalf:/files]# cat
test_with_consistency_check
Tue Aug 12 14:59:54 CEST 2008Now we go back to the server, and append additional data to the file:
[root@theoden:/export/files]# date >> test_with_consistency_check
[root@theoden:/export/files]# cat test_with_consistency_check
Tue Aug 12 14:59:54 CEST 2008
Tue Aug 12 15:00:11 CEST 2008And obviously, you will see this change on the client:[root@gandalf:/files]# cat
test_with_consistency_check
Tue Aug 12 14:59:54 CEST 2008
Tue Aug 12 15:00:11 CEST 2008
```

With on-demand consistency checkNow we unmount it, and remount it:

```
[root@gandalf:/files]# cd /
[root@gandalf:/]# umount /files
[root@gandalf:/]# mount -F cachefs -o backfstype=nfs,backpath=/var/cachefs/backpaths/files,
cachedir=/var/cachefs/caches/cache1,demandconst theoden:/export/files /files You may have noticed the demandconst
option. This option changes everything.Let's assume you created another file on the NFS server:
[root@theoden:/export/files]# date >> test_with_ondemand_consistency_check
[root@theoden:/export/files]# cat test_with_ondemand_consistency_check
Tue Aug 12 15:00:57 CEST 2008Back on the NFS client you will not even see this file:
[root@gandalf:/files]# ls
index.html                pcre_refcount.html
[...]
pcre_info.html            pcretest.html
pcre_maketables.html     test_with_consistency_checkYou have to trigger a consistency check. This is quite easy.
```

```
[root@gandalf:/files]# cfsadmin -s all
```

Now you can see the file in the directory.

```
[root@gandalf:/files]# ls
index.html                pcre_study.html
[.]
pcre_info.html           test_with_consistency_check
pcre_maketables.html    test_with_ondemand_consistency_check
pcre_refcount.htmlOkay, now we can look into the file.[root@gandalf:/files]cat test_with_ondemand_consistency_check
Tue Aug 12 15:00:57 CEST 2008Now we append a new line to the file on the server by executing the following
commands on the NFS server
[root@theoden:/export/files]date >> test_with_ondemand_consistency_check
[root@theoden:/export/files]cat test_with_ondemand_consistency_check
Tue Aug 12 15:00:57 CEST 2008
Tue Aug 12 15:02:03 CEST 2008When we check this file on our NFS client, we still see the cached
version.[root@gandalf:/files]cat test_with_ondemand_consistency_check
```

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

Tue Aug 12 15:00:57 CEST 2008 So let's trigger a consistency check: [root@gandalf:/files] cfsadmin -s all Now we can look into the file again, and you will see the new version of the file.

```
[root@gandalf:/files] cat test_with_ondemand_consistency_check
```

Tue Aug 12 15:00:57 CEST 2008

Tue Aug 12 15:02:03 CEST 2008 Okay, it's pretty obvious this isn't a feature for a filesystem that change in a constant and fast manner. But it's really useful for situations, where you have control over the changes. As long as a file is cached, the file server will see not a single access for such files. Thus such a file access doesn't add to the load of the server.

There is an important fact here: It doesn't tell CacheFS to check the files right at that moment. It just tells CacheFS to check it at the next access to the file. So you don't have an consistency check storm.

Posted by Joerg Moellenkamp in English, Solaris at 14:47