

Monday, August 18, 2008

Less known Solaris Features: CacheFS - Part 3: A basic example

Okay ... using CacheFS is really easy. Let's assume, you have an fileserver called theoden We use the directory /export/files as the directory shared by NFS. The client in our example is gandalf.
PreparationsLet's create a NFS server at first. This is easy. Just share an directory on a Solaris Server. We login onto theoden and execute the following commands with root privileges.

```
[root@theoden:~]# mkdir /export/files
[root@theoden:~]# share -o rw /export/files
# share
- /export/files rw ""Okay, of course it would be nice to have some files to play around in this directory. I will use some files of the Solaris Environment.[root@theoden:~]# cd /export/files
[root@theoden:/export/files]# cp -R /usr/share/doc/pcre/html/* .Let's do a quick test, if we can mount the directory:[root@gandalf:~]# mkdir /files
[root@gandalf:~]# mount theoden:/export/files /files
[root@gandalf:~]# unmount /filesNow you should be able to access the \verb=/export/files= directory on theoden by accessing \verb=/files= on gandalf. There should be no error messages.
```

Okay, at first we have to create the location for our caching directories. Let's assume we want to place our cache at /var/cache/fs/caches/cache1. At first we create the directories above the cache directory. You don't create the last part of the directory structure manually.[root@gandalf:~]# mkdir -p /var/cache/fs/cachesThis directory will be the place where we store our caches for CacheFS. After this step we have to create the cache for the CacheFS.[root@gandalf:/files]# cfsadmin -c -o maxblocks=60,minblocks=40,threshblocks=50 /var/cache/fs/caches/cache1The directory cache1 is created automatically by the command. In the case the directory already exists, the command will quit without creating the cache.

Additionally you have created the cache and you specified some basic parameters to control the behaviour of the cache. Citing the manpage of cfsadmin:

maxblocks: Maximum amount of storage space that CacheFS can use, expressed as a percentage of the total number of blocks in the front file system.

minblocks Minimum amount of storage space, expressed as a percentage of the total number of blocks in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms.

threshblocks A percentage of the total blocks in the front file system beyond which CacheFS cannot claim resources once its block usage has reached the level specified by minblocks.

All this parameter can be tuned to prevent CacheFS to eat away all the storage available in a filesystem, a behaviour that was quite common to early versions of this feature.

Mounting a filesystem via CacheFSWe have to mount the original filesystem now.[root@gandalf:/files]# mkdir -p /var/cache/fs/backpaths/files

```
[root@gandalf:/files]# mount -o vers=3 theoden:/export/files /var/cache/fs/backpaths/filesYou may have noticed the parameter that sets the NFS version to 3. This is necessary, as CacheFS isn't supported with NFSv4. Thus you can only use it with NFSv3 and below. The reason of this limitation has its foundation in the different way NFSv4 handles inodes.
```

Okay, now we mount the cache filesystem at the old location:[root@gandalf:/files]# mount -F cache fs -o backfstype=nfs,backpath=/var/cache/fs/backpaths/files,cachedir=/var/cache/fs/caches/cache1 theoden:/export/files /filesThe options of the mount controls some basic parameters of the mount:

backfstype specifies what type of filesystem is proxied by the CacheFS filesystem

backpath specifies where this proxied filesystem is currently mounted

cachedir specifies the cache directory for this instance of the cache. Multiple CacheFS mounts can use the same cache.

From now on every access to the /files directory will be cached by CacheFS. Let's have a quick look into the /etc/mnttab. There are two important mounts for us:[root@gandalf:/etc]# cat mnttab

```
[...]
theoden:/export/files /var/cache/fs/backpaths/files nfs vers=3,xattr,dev=4f80001 1219049560
/var/cache/fs/backpaths/files /files cache fs
backfstype=nfs,backpath=/var/cache/fs/backpaths/files,cachedir=/var/cache/fs/caches/cache1,dev=4fc0001
```

1219049688The first mount is our back file system, it's a normal NFS mountpoint. But the second mount is a special one. This one is the consequence of the mount with the `\verb=-F cachefs=` option.

Statistics about the cacheWhile using it, you will see the cache structure at `/var/cachefs/caches/cache1` filling up with files. I will explain some of the structure in the next section. But how efficient is this cache? Solaris provides an command to gather some statistics about the cache. With `cachefsstat` you print out data like hit rate inclusive the absolute number of cache hits and cache misses:`[root@gandalf:/files]# /usr/bin/cachefsstat`

```
/files
  cache hit rate: 60% (3 hits, 2 misses)
  consistency checks: 7 (7 pass, 0 fail)
    modifies: 0
  garbage collection: 0
[root@gandalf:/files]#
```

Posted by Joerg Moellenkamp in English, Solaris at 12:59

\verb shows up in "Preparations" section - looks like a page format error. Also lots of grammaticos e.g. "Just share an directory"
Anonymous on Sep 29 2008, 21:04