Unit VII – more realistic SRM scenarios

preparing the VM

- revert to the snapshot to undo all previous exercises/dCache configurations
- OR just delete your existing VM and download a fresh (and empty) copy from

```
http://trac.dcache.org/trac.cgi/wiki/dCacheToGo
```

setup a T0D1 area (one physical pool)

- login as ROOT user into your VM
- enable SpaceManagement (if not done before) in */opt/d-cache/config/dCacheSetup*

```
# enable global space management
srmSpaceManagerEnabled=yes

# allow implicit spacemanagement w/o space tokens
srmImplicitSpaceManagerEnabled=yes
```

  restart the SRM component:

```
dcache restart srm
```

- create a link group reflecting T0D1 (aka Replica/Online) in the PoolManager

  login to the *PoolManager* via the ssh-interface

```
ssh -c blowfish -p 22223 -l admin hal9000
cd PoolManager

# create a new link group
psu create linkGroup T0D1-link-group

# list the (default) properties of this link group
psu ls linkGroup -l
T0D1-link-group : [EMPTY]
     Attributes:
     AccessLatency:
          onlineAllowed=false
          nearlineAllowed=true
     RetentionPolicy:
          custodialAllowed=true
          outputAllowed=true
          replicaAllowed=true
```

```
# we don't want anything else then Replica/Online
psu set linkGroup onlineAllowed       T0D1-link-group true
psu set linkGroup replicaAllowed      T0D1-link-group true
psu set linkGroup nearlineAllowed     T0D1-link-group false
psu set linkGroup custodialAllowed    T0D1-link-group false
psu set linkGroup outputAllowed       T0D1-link-group false

# list existing links (only default-link should exist)
psu ls link

# create a new link
psu create link T0D1-link any-protocol any-store world-net

# set link preferences (read/write, no pool2pool and no stage)
psu set link T0D1-link -readpref=20 -writepref=20
     cachepref=0 -p2ppref=0

# get a list of pools available (should return 4 pools)
psu ls pool

# move the target pool from default poolgroup to the link
psu removefrom pgroup default hal9000_1
psu add link T0D1-link hal9000_1

# add the link to the link group
psu addto linkGroup T0D1-link-group T0D1-link

# check your setup
psu ls pool hal9000 -l
psu ls link T0D1-link -l
psu ls linkGroup T0D1-link-group -l
free

# persist your setup
save

# exit the dCache shell
..
logoff
```

- authorize the link group

  set the location of the authorization setup file in */opt/d-cache/config/dCacheSetup*

```
SpaceManagerLinkGroupAuthorizationFileName=
     /opt/d-cache/etc/LinkGroupAuthorization.conf
```

  now edit */opt/d-cache/etc/LinkGroupAuthorization.conf* and allow the reservation of space for all members of VO 'desy':

```
LinkGroup T0D1-link-group
```

```
/desy/Role=*
```

- login back into the dCache shell and enter the SrmSpaceManager

```
cd SrmSpaceManager
```

pick up the changes made to the linkGroup authorization file

```
update link groups
```

query for your link group

```
ls -l

LinkGroups:0 Name:T0D1-link-group FreeSpace:523055504
ReservedSpace:0 AvailableSpace:523055504  .. ..
```

- setup implicit space reservation

make sure the Chimera namespace is mounted:

```
mount localhost:/pnfs /pnfs
```

create the new target directory and set the ownership

```
mkdir /pnfs/dcache.org/data/DISK
chown desy01:desy /pnfs/dcache.org/data/DISK/
```

Only users from VO 'desy' can now write into this directory, because they get all mapped to the local user 'desy01' as configured in the gPlazma setup.

Now set the default AccessLatency/RetentionPolicy for this directory

```
echo "REPLICA" >
     /pnfs/dcache.org/data/DISK/".(tag)(RetentionPolicy)"

echo "ONLINE" >
     /pnfs/dcache.org/data/DISK/".(tag)(AccessLatency)"
```

Check your settings

```
cat /pnfs/dcache.org/data/DISK/".(tag)(RetentionPolicy)"
cat /pnfs/dcache.org/data/DISK/".(tag)(AccessLatency)"
```

- write into this space

login as “ui_user” into your VM and obtain a *voms-proxy*:

```
voms-proxy-init --voms desy
```

write into your space simply by selecting the directory:

```
srmcp -2 file:////bin/sh
     srm://hal9000:8443/pnfs/dcache.org/data/DISK/file1
```

This works because the directory */pnfs/dcache.org/data/DISK* is directly connected to the T0D1 linkgroup via the tags we set. You can of course write to another directory (accounted in the same linkgroup), but then you need to specify Accesslatency/RetentionPolicy:

```
srmcp -2 -access_latency=ONLINE -retention_policy=REPLICA
     file:////bin/sh
     srm://hal9000:8443/pnfs/dcache.org/data/file2
```

Right now, your dCache instance is partitioned in one SRM managed area (T0D1, attached to pool hal9000_1) and the unmanaged area (the pools hal9000_2, _3 and _4).

Attaching a HSM (aka tape backend) to pools

This section describes how to connect some of the pools to an external script, which makes dCache believe a tape backend is connected. We will do that only for pools hal9000_2, hal9000_3 and hal9000_4, since hal9000_1 is already assigned to a T0D1 area.

- become ROOT in your VM and backup the currentpool setup

```
cp /pools/2/pool/setup /pools/2/pool/setup.backup
cp /pools/3/pool/setup /pools/3/pool/setup.backup
cp /pools/4/pool/setup /pools/4/pool/setup.backup
```

- install the HSM setup for these pools

```
cp /pools/2/pool/setup.fakeHSM /pools/2/pool/setup
cp /pools/3/pool/setup.fakeHSM /pools/3/pool/setup
cp /pools/4/pool/setup.fakeHSM /pools/4/pool/setup
```

- restart the pools

```
dcache restart pool
```

- test the (simulated) flush/stage

  First of all, write a file to the *unmanaged* space in order to delegate the file to pool 2,3 or 4. Become 'ui_user', get a *voms-proxy* and then do:

```
globus-url-copy file:////bin/sh
```

```
gsiftp://hal9000:2811/pnfs/dcache.org/data/hsmFile1
```

Login to the ssh-interface and lookup the pool where the previously written file resides on:

```
cd PnfsManager

cacheinfoof /pnfs/dcache.org/data/hsmFile1
hal9000_3

pnfsidof /pnfs/dcache.org/data/hsmFile1
0000895F5DB4DC374CACBC7C92D324D62FCF
```

Go to the pool shown above and check the file status with the pnfsid you just got:

```
..
cd hal9000_3
rep ls 0000895F5DB4DC374CACBC7C92D324D62FCF
0000895F5DB4DC374CACBC7C92D324D62FCF <-P---------
(0)[0]> 616248 si={sql:chimera}
```

The flag <*P*> means the file is in state "precious" and is ready to go to tape. Trigger the actual flush to the (simulated) HSM by doing:

```
flush pnfsid 0000895F5DB4DC374CACBC7C92D324D62FCF
```

Check the file status again, it should display file status <*C*> which means "cached" or "copied to tape, still readable from the outside world, but might be removed as soon as pool runs short on space".

```
rep ls 0000895F5DB4DC374CACBC7C92D324D62FCF
0000895F5DB4DC374CACBC7C92D324D62FCF <C--------->
```

A 2nd copy of the file is now stored on tape (not in our simulated case), so dCache removes the (least recently used) file from the pool if the space is needed for new, arriving files.

Just trigger the removal from the pool (and therefore from disk) by

```
rep rm 0000895F5DB4DC374CACBC7C92D324D62FCF
```

Now stage the file back into the pool by

```
rh restore 0000895F5DB4DC374CACBC7C92D324D62FCF
```

Please note that the simulating HSM script used in this VM actually not copies the file to another place, it just pretends to do so to satisfy dCache. On a stage operation, it will just

generate a new file with the particular filesize, but filled with Zeros only. So don't expect any useful data after a stage ;)

We now successfully tested the manual flush/stage of a file. Let's move on to automate this (as you would do in a production system):

Exit the dCache shell and edit again the pool setup files:

```
/pools/2/pool/setup
/pools/3/pool/setup
/pools/4/pool/setup
```

Uncomment the following line in each of those files:

```
pool lfs none
```

After restarting the pools, any precious files should be flushed automatically and marked as "cached". To enable auto-staging on a user's file-request, add the following line to */opt/d-cache/config/PoolManager.conf* :

```
rc set stage on
```

Now restart the poolmanager that make the changes take affect:

```
dcache restart dcache
```

Now we can test the full lifecycle of a custodial file:

- write a file into dCache (HSM-connected pools), e.g. using *gsiFtp*
- remove the (cached) file manually from the pool
- read the file from back from dCache with gsiFtp.
  It will be staged on demand onto one of the pools 2-4 and then served to the client.
- optional: login to the PnfsManager and find the new pool where the file was staged on

Creation of a T1D1 area (one physical pool connected to a "HSM")

- setup a new link group in the PoolManager via the ssh-interface:

```
cd PoolManager

psu create linkGroup T1D1-link-group

# set attributes ONLINE/CUSTODIAL (aka T1D1)
psu set linkGroup onlineAllowed T1D1-link-group true
psu set linkGroup custodialAllowed T1D1-link-group true
```

```
psu set linkGroup nearlineAllowed T1D1-link-group false
psu set linkGroup replicaAllowed T1D1-link-group false
psu set linkGroup outputAllowed T1D1-link-group false

# create a appropriate link, which allows read, write and stage
psu create link T1D1-link any-protocol any-store world-net
psu set link T1D1-link -readpref=20 -writepref=20 -cachepref=20


# move pool hal9000_2 from the default poolgroup into the new
# link
psu removefrom pgroup default hal9000_2
psu add link T1D1-link hal9000_2
psu addto linkGroup T1D1-link-group T1D1-link

# check your setup
psu ls pool hal9000_2 -l
psu ls link T1D1-link -l
psu ls linkGroup T1D1-link-group -l
free

# persist configuration and exit the shell
save
..
logoff
```

- authorize the link group created above

  edit /opt/d-cache/etc/LinkGroupAuthorization.conf (all DESY VO members authorized) and add the lines:

  ```
  LinkGroup T1D1-link-group
  /desy/Role=*
  ```

- create a new directory in the namespace to separate T1D1 files

  ```
  mkdir /pnfs/dcache.org/data/DISKTAPE
  ```

  writable only by 'desy01', where all DESY members are mapped to

  ```
  chown desy01:desy /pnfs/dcache.org/data/DISKTAPE
  ```

- login back to the dCache ssh-interface and go visit the SrmSpaceManager

  ```
  cd SrmSpaceManager
  ```

  reserve a space:

  ```
  reserve -vog=/desy -vor=NULL -acclat=ONLINE -retpol=CUSTODIAL
  -desc=T1D1 -lg=T1D1-link-group 50MB "-1"
       10000 .. ..
  ```

get your space token details by:

```
ls -l 10000
Reservations:
     10000 voGroup:/desy .. ..
```

- try to write a file into the token

```
srmcp -2 -space_token=10000 file:////bin/sh
srm://hal9000:8443/pnfs/dcache.org/data/DISKTAPE/file1
```

login back to the ssh interface of dCache and go to pool hal9000_2:

```
cd hal9000_2
rep ls
```

there should be the following flags displayed for the file:

```
rep ls
0000895F5DB4DC374CACBC7C92D324D62FCF <C-------X--(0)[0]> 616248
     si={sql:chimera}
```

This means, that the file went already to tape but is pinned on disk forever, reflecting the T1D1 characteristics.

Creation of a T0D1 area (one write-pool, one read-pool)

- create a new directory

```
mkdir /pnfs/dcache.org/data/TAPE
```

Make it writeable only for the production user "/desy/Role=Production", which is mapped to local user desy02

```
chown desy02:desy /pnfs/dcache.org/data/TAPE
ls -ld /pnfs/dcache.org/data/TAPE
```

- create a new link group

```
cd PoolManager

psu create linkGroup T1D0-link-group

# set attributes CUSTODIAL/NEARLINE (aka T0D1)
psu set linkGroup nearlineAllowed T1D0-link-group true
psu set linkGroup custodialAllowed T1D0-link-group true
psu set linkGroup onlineAllowed T1D0-link-group false
psu set linkGroup replicaAllowed T1D0-link-group false
psu set linkGroup outputAllowed T1D0-link-group false
```

```
# create a write-only link
psu create link T1D0-write-link any-protocol any-store world-net

psu set link T1D0-write-link -readpref=0 -writepref=20
-cachepref=0

# assign a pool to the write-only link
# (acting as a temporary diskbuffer in front of the tape)
psu removefrom pgroup default hal9000_3
psu add link T1D0-write-link hal9000_3
psu addto linkGroup T1D0-link-group T1D0-write-link


# create another link which is used only for staging and reading
# and is not part of any linkgroup
psu create link T1D0-read-link  any-protocol any-store world-net
psu set link T1D0-read-link -readpref=20 -writepref=0
-cachepref=20

# move a pool from the default linkgroup into the read-link
psu removefrom pgroup default hal9000_4
psu add link T1D0-read-link hal9000_4

# check your setup
psu ls pool hal9000_3 -l
psu ls pool hal9000_4 -l
psu ls link T1D0-write-link -l
psu ls link T1D0-read-link -l
free
```

note that the write-link is part of the linkGroup, but the read-link is not

```
psu ls linkGroup T1D0-link-group -l


save
```

- authorize the linkGroup in /opt/d-cache/etc/LinkGroupAuthorization.conf by adding the lines:

```
LinkGroup T1D0-link-group
/desy/Role=production
```

- manage the linkgroup in the SpaceManager

go back to SpaceManager in the dCache ssh-interface

```
cd SrmSpaceManager
```

pick up the new link group (if not done already)

```
update link groups
```

check that the new linkGroup is known to the SpaceManager and note that only "/desy/Role=prodcution" is authorized

```
ls -lg=T1D0-link-group
30000 Name:T1D0-link-group FreeSpace:524288000 ReservedSpace:0
AvailableSpace:524288000 VOs:{/desy:production} .. ..
```

- Write into this linkGroup

  First, you need to obtain a new proxy with the role "production"

```
voms-proxy-init  -- voms  desy:/desy/Role=production
```

  Go on an write a file. Note that no space token nor AccessLatency/RetentionPolicy needs to be defined. Do you know why?

```
srmcp -debug -2 file:////bin/sh
      srm://hal9000:8443/pnfs/dcache.org/data/TAPE/hsmFile1
```

  If you are fast, you can see the free space of the linkgroup T1D0-link-group being shrinked. But as soon as the file is flushed to tape, the linkgroup free space should increase again. This happens due to the WLCG agreement that T0D1 does only take the disk space into account, not the space used on tape (Actually, tape is considered to be "infinite").

- check the status of the file within the SrmSpaceManager

```
cd SrmSpaceManager
```

  find out the (implicit) space token where this file lives in

```
ls file space tokens  /pnfs/dcache.org/data/TAPE/hsmFile1
30009 /desy production .. ..
```

  check the status of the file

```
listFilesInSpace 30009
.. .. /pnfs/dcache.org/data/TAPE/hsmFile1
 0000E545700FE4B44A578AE238B61E5BDB11 Flushed 0
```

  ("Stored" means file is not yet flushed to tape, "Flushed" means file is copied to tape)

  If the state of the file is "Flushed", the free space of the linkGroup should have increased again by the filesize of /pnfs/dcache.org/data/TAPE/hsmFile1

  Feel free to reserve explicit spaces via

```
reserve -vog=/desy -vor=production -acclat=NEARLINE
-retpol=CUSTODIAL -desc=T1D0 -lg=T0D1-link-group  <some size>
"-1"
```

Note that only the "desy" user with role 'production' is allowed to write into this T1D0 area as defined by us. This is enforced by the Unix file permissions on /pnfs/dcache.org/data/TAPE. Since that directory is group-readable, the "normal" desy VO-member can at least read.