

Handout Unit IV – recovering from hanging file transfers

Simulate failing pools (on your PoolNode)

```
/opt/d-cache/bin/dcache-pool stop
```

Try to read a file from the UI which is stored on a “failing” pool (should hang)

```
globus-url-copy -dbg gsiftp://<yourHeadNode>.desy.de:2811/..  
file:///dev/null
```

Go through the checklist

- connect to the ssh-interface of dCache and find out the pnfsid of your file you tried to read

```
cd PnfsManager  
pnfsidof <PathOfYourFailingFile>
```

- go to the PoolManager and list current (suspended) requests

```
cd PoolManager  
rc ls
```

the last command should have returned a list of requested pnfsids, as well as the cause for the error (“Suspended – pool is disabled”). You now know officially that the transfer hangs because it's pool is not available to the PoolManager

- try to cd into that pool

```
..  
cd <pool>
```

the last command should return “No route to cell”. Now you know that the pool is really down.

- login to your PoolNode and restart the pool (normally you would check the pool log file to get an idea what went wrong)

```
/opt/d-cache/bin/dcache-pool start
```

All hanging requests (no matter of whether it was srm, dcap or ftp) should recover and go on with the transfer automatically!

Retry your read-request in case auto-recovery didn't work (from the UI using globus-url-copy)

- if it's still not working, login again to the ssh-interface of dCache

```
cd PoolManager  
rc ls
```

the pnfsid should be still listed there, meaning that the request is still suspended, no matter how often you retry to read the file. 'Suspended' means to the PoolManager an unexpected error occurred while executing this request and human interaction is required.

- solve the problem by deleting the suspended request

```
rc destroy <pnfsID> - */*
```

- retry the globus-url-copy, should work now