Goal of Unit I

- complete all-in-one-node dCache instance installed via YAIM
  - the so called “headnode”
  - PoolManager, SRM, Infoprovider, 2 pools, various doors (dcap, gsidcap, gsiftp, xrootd)
  - one VO supported (dteam) an your 1st DN authorized
- testing your instance
  - basic transfers using srmcp, globus-url-copy, dccp
- getting in touch with administration
  - web- and ssh-interface, billing

Getting started

- preparation (see handout for details)
    - login from your laptop to the UI (1st account)
    - from there you should have password-less root access to your personal VM (the first in the list)
- on your personal VM you will find:
    - “naked” SL4, 512 MB RAM
    - host certificate in /etc/grid-security
    - yum repositories are already setup
        - official scientific linux repos
        - Cern CA repo
        - Cern Glite 3.1 repos
        - dCache.org repos (latest versions of dCache, YAIM)

Installation

- yum install the following packages
  - glite-SE_dcache_admin_postgres
  - glite-SE_dcache_info
  - lcg-CA
  - lcg-vomscerts
  - glite-yaim-dcache
  - glite-version
  - compat-postgresql-libs

core dCache
(headnode +
infoprovider)

the LCG
CA certificates

the YAIM
installer

additional
dependencies

Special steps required by workshop environment

- each group got 2 test certificates, which are unknown to the VOMS server
  -> no VOMS-based authz during workshop
- setting up dCache authorization without VOMS requires following steps
  - install additional, gridmap-specific RPMs
  - YAIM merges the testusers (YOU from the local gridmap file) and the offical dteam-users into a global gridmap-file and a dCache kpwd-file
  - dCache (gPlazma) will be made aware of new kpwd file

    -> you don't have to do all this with certificates of real users signed by real CAs

Configuration

- take the default schema for local users and groups from the YAIM examples
- setup site-info.def based on the template

Your site-info.def

```
MY_DOMAIN=desy.de

JAVA_LOCATION="<yourJDKHomePath>"

DCACHE_ADMIN="<yourHeadNode>.desy.de"

DCACHE_POOLS="<yourHeadNode>.desy.de:7:/pools/1 <yourHeadNode>.desy.de:7:

DCACHE_DOOR_SRM="<yourHeadNode>.desy.de"
DCACHE_DOOR_GSIFTP="<yourHeadNode>.desy.de"
DCACHE_DOOR_GSIDCAP="<yourHeadNode>.desy.de"
DCACHE_DOOR_DCAP="<yourHeadNode>.desy.de"
DCACHE_DOOR_XROOTD="<yourHeadNode>.desy.de"
DCACHE_DOOR_LDAP="<yourHeadNode>.desy.de"

RESET_DCACHE_CONFIGURATION=yes
RESET_DCACHE_PNFS=yes
RESET_DCACHE_RDBMS=yes

VOS="dteam"
```

Home of JDK, NOT JRE

specifies the headnode

2 pools on the headnode

all kinds of doors, SRM

the infoprovider host

recreate dCache conf files, databases and PNFS

we only support one VO so far

Run YAIM

- run YAIM to configure and start dCache, based on your site-info.def
    - this may a few minutes..
- as soon as YAIM completes, you should have a complete dCache instance up and running !!

- edit /etc/grid-security/storage-authzdb

```
# authzdb for dteam001 added by dcacheVoms2Gplasma
authorize dteam001 read-write 18118 2688 / / /

# authzdb for sgmdtm01 added by dcacheVoms2Gplasma
authorize sgmdtm01 read-write 60501 2690 / / /

# authzdb for prddtm01 added by dcacheVoms2Gplasma
authorize prddtm01 read-write 50501 2689 / / /
```

- restart core-dCache:
    - /opt/d-cache/bin/dcache-core restart

Check what is running

- check in the CLI that all Domains (=java processes) are up
- check which components (=cells) registering themselves in the web-interface
- try filesystem-commands on the exported PNFS namespace (like cd, ls)
- don't cp directly into PNFS, the file then goes as a BLOB into the database!!!

test basic transfers

- local Dcap transfer into dCache, using the exported (mounted) PNFS namespace
- from the UI
    - plain dcap, gsiDcap, GridFtp, SRM copy
- plain dCap should fail from the UI

  Why?

The ssh-interface to dCache

- login into dCache

- exercise:

  find the pool where a certain file resides on (a file which you have written into your dCache before)