dCache Installation and Quattor

Stijn De Weirdt
IIHE, VUB

dCache workshop II 18-19/01/07

dCache installation and Quattor: overview

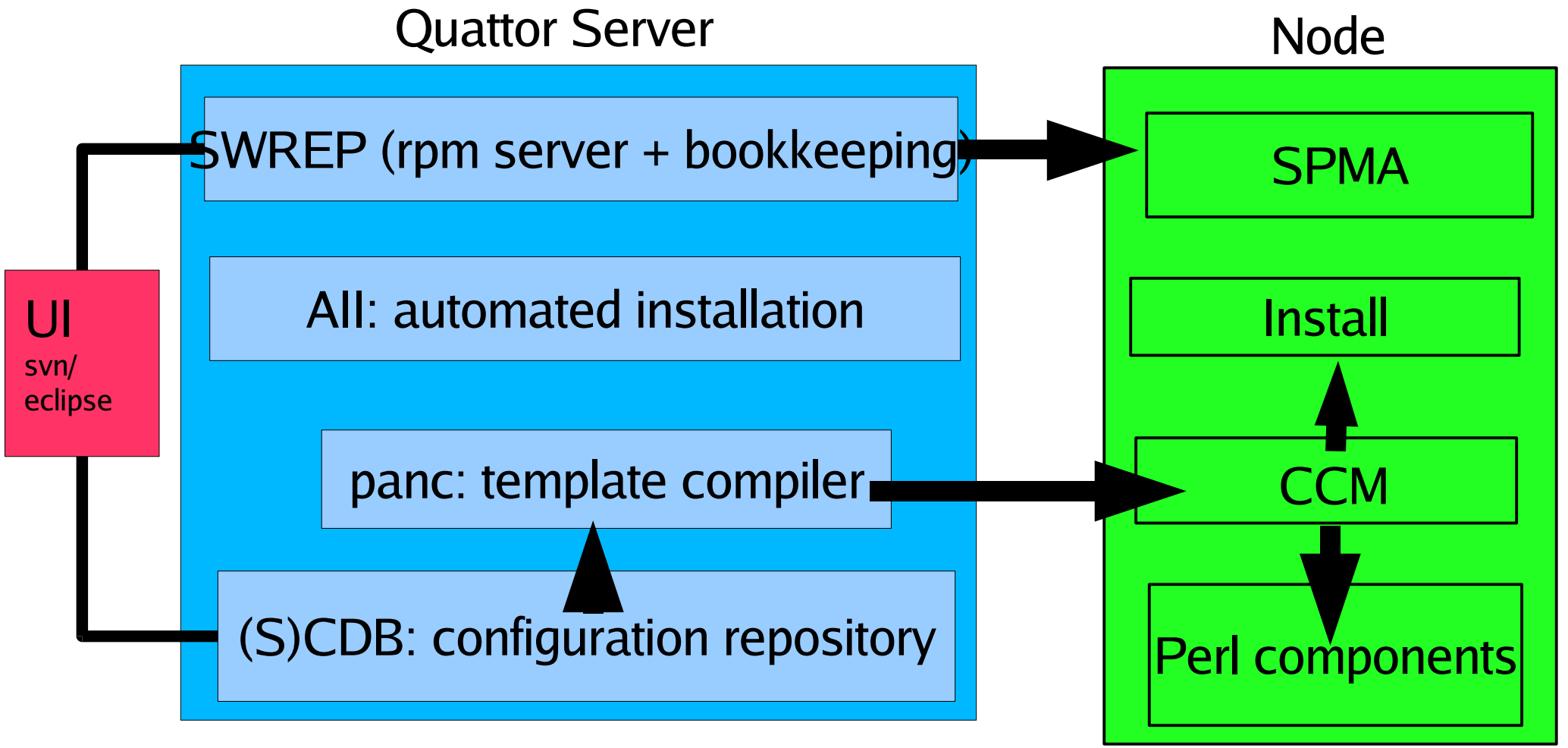
- Quattor overview
  - Quattor layout
  - Grid deployment mode
  - Pro/contra
- Ncm-dcache
  - Component description
  - Supported features
  - Component structure
  - Pro/contra
  - Examples
- Future

dCache installation and Quattor: Quattor overview

- Quattor is an Administration ToolkiT for Optimizing Resources
- Fabric management system for installation, configuration and management
  - Autonomous nodes
    - Local config scripts
    - No remote managment
  - Central configuration
    - Single source of information
    - Version control
  - Reproducibility
    - Idempotent operators
  - Scalability
    - Not only for farms
    -

- More info on http://quattor.org

dCache installation and Quattor: component overview



dCache installation and Quattor: quattor server

- SCDB: node configuration described by pan
  - Template: description of a part of the config
  - Panc: pan compiler generates xml profile
    - Declarative node description
  - Accessible by http/https
  - Subversion based
    - CDB: cvs based, for installation with high user interaction
- SWREP: stores and keeps track of RPMs
  - http/nfs access
- AII: configures and deals with the installation
  - ks-files, DHCP config, PXE, rescue mode

dCache installation and Quattor: quattor node

CCM: Configuration Cache Manager

- fetches profile, keeps local copy
- listens for configuration changes
- provides the interface to the components

NCM: Node Configuration Manager

- checks if configuration = actual state

Components: do the actual configuration work

- perl modules interacting on local OS
- ncm-<something>

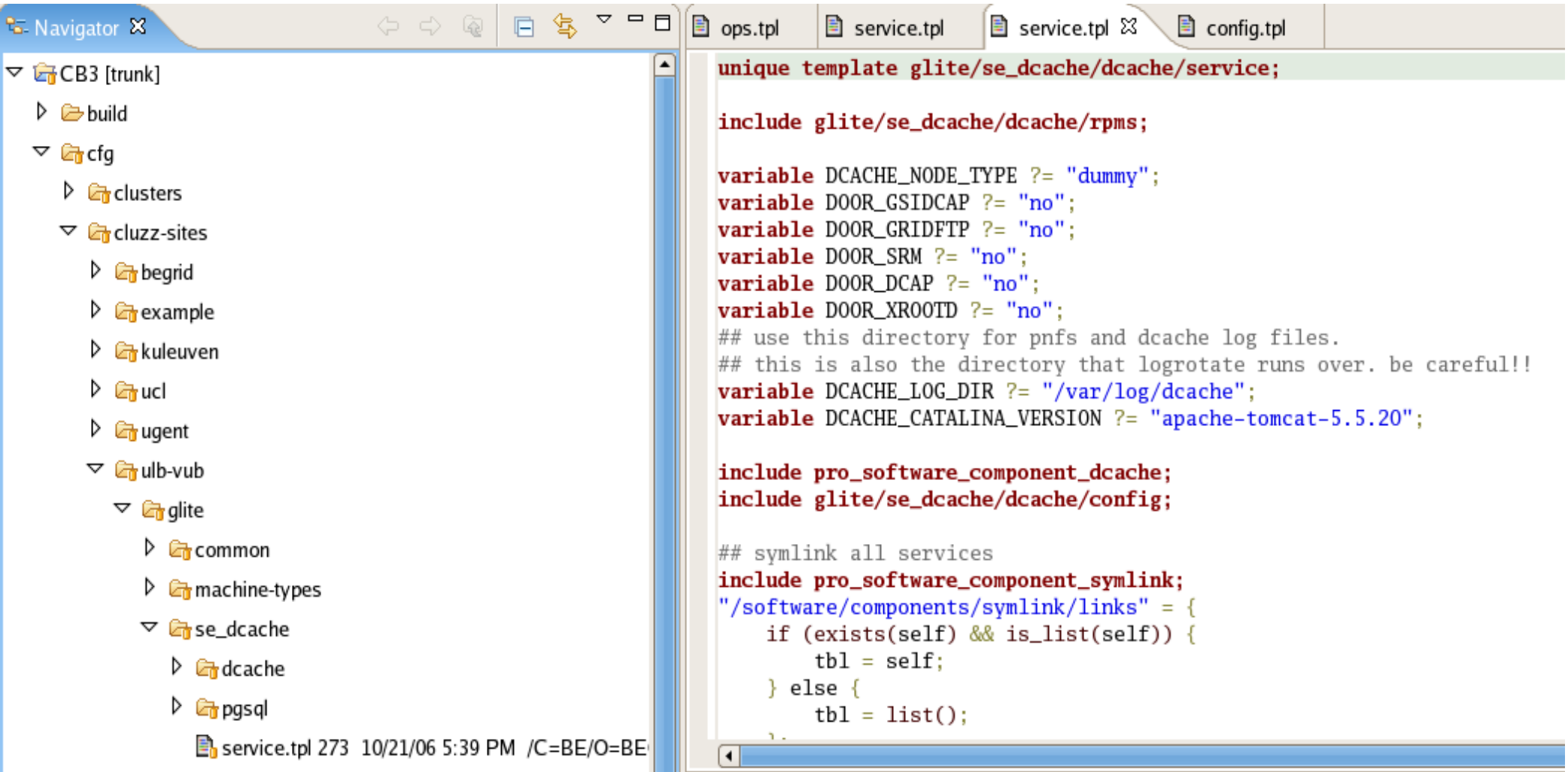
SPMA: special component for the software

- Supports “user-mode” (desktop support)
- Quattor can be used with replacing SPMA with apt (but do you want this?)

dCache installation and Quattor: Quattor UI to configuration

eg SCDB:

- Standard svn command line client + ant
- Something more visual: eclipse + syntax highlighting + ant



dCache installation and Quattor: Quattor grid setup

2 modes

- YAIM based: quattor yaim component generates YAIM config file, runs YAIM scripts
- QWG based: manages grid services as any other component
  - Needs dedicated components
  - One gLite component for all gLite services
  - Manpower issues, delayed releases
  - But stable

dCache installation and Quattor: Quattor pro/contra

Advantages

- Share configuration work
- Easy to manage multiple (similar) sites with very few people
  - GridIreland, GRIF, BEgrid, (CERN)
- One tool for a lot of things

Disadvantages

- Learning curve
- Rpm configuration
  - No “apt” style generation of rpm lists (yet)
- QWG release delays
- User base is not so big

dCache installation and Quattor: ncm-dcache

ncm-dcache : dcache Quattor component

- Try to automate the boring/repetitive tasks
  - Eg add a new pool node without really knowing what dcache does
- Perl
  - Written in beginner-mode, very readable ;)
- Modular
  - Should be easy to add features
- Very careful, a lot of output
- ncm-postgresql
  - Split off from ncm-dcache
  - Very basic options

dCache installation and Quattor: ncm-dcache (II)

What works?

- 1.6.6-5, 1.7.0 (not yet out)
- Install dcache admin/pool/doors
  - Split admin node is untested
- PNFS: modify pnfs_setup, add pnfsdatabases, set export rules
- Modify dCacheSetup and node_config files
- Pools: size(dyn/stat), create/add to pgroup
- Configure/create link,unit/ugroup
- TODO: queues, more 170 features

dCache installation and Quattor: component structure

0. Postgres is running/startable (ncm-postgres does that)
1. Read quattor config, compare with current setup and then do something
2. Check basics, explicit version check, modify scripts
   - Is nfs module loaded?, is it 1.6.6 or 1.7.0?, add ulimit -n to dcache-pool script
   - Fixes for problems that once occured
3. First time install?
4. Further configuration
   - Try to only modify what is really needed
   - Config files using templates, if available
   - Most is through admin interface
   - Restarts if needed
5. (generation of slony scripts)

dCache installation and Quattor: ncm-dcache pro/contra

Advantages

- Easy and flexible
- Central configuration of sites (see example configuration next slide)
  - Configuration of one machine can be based on all available site info
- Easy to extent for admin interface configuration (see example code next slide)
  - Write regexp to catch useful info
  - Write generator for admin interface commands

Disadvantages

- Maintainer
  - no standard dcache tools/scripts for this
  - Testing is limited, not all features supported
- Yet Another dCache Manager
- Central approach: if something is missing, you will have to retrigger the install/configuration on other depending nodes

dCache installation and Quattor: ncm-dcache example configuration

```
"/software/components/dcache/pool/ignore_pgroup" = list("default");
"/software/components/dcache/pool/pools"=nlist(
    ## out_buf pgroup: for outside buffer reading and writing
    #"behar",list(nlist("path","/storage/1","pgroup",merge(CMS_VOS,list("out_buf")),"mover_max","1000"),
    #             nlist("path","/storage/2","pgroup",merge(CMS_VOS,list("out_buf")),"mover_max","1000"),
    #             nlist("path","/storage/3","pgroup",merge(CMS_VOS,list("out_buf")),"mover_max","1000")),
    "behar",list(nlist("path","/storage/1","pgroup",merge(CMS_VOS,list("out_buf","behar_test")),"mover_max","1000"),
                 nlist("path","/storage/2","pgroup",merge(CMS_VOS,list("out_buf","behar_test")),"mover_max","1000"),
                 nlist("path","/storage/3","pgroup",merge(list("behar_test")),"mover_max","1000")),
    "behar2",list(nlist("path","/storage/1","pgroup",merge(CMS_VOS,list("out_buf","behar2_test")),"mover_max","1000"),
                 nlist("path","/storage/2","pgroup",merge(CMS_VOS,list("out_buf","behar2_test")),"mover_max","1000"),
                 nlist("path","/storage/3","pgroup",merge(CMS_VOS,list("out_buf","behar2_test")),"mover_max","1000")),
    "behar3",list(nlist("path","/storage/1","pgroup",merge(CMS_VOS,list("out_buf","behar3_test")),"mover_max","1000"),
                 nlist("path","/storage/2","pgroup",merge(CMS_VOS,list("out_buf","behar3_test")),"mover_max","1000"),
                 nlist("path","/storage/3","pgroup",merge(CMS_VOS,list("out_buf","behar3_test")),"mover_max","1000")),
    "behar5",list(nlist("path","/storage/1","pgroup",merge(CMS_VOS,list("out_buf","behar5_test")),"mover_max","1000"),
                 nlist("path","/storage/2","pgroup",merge(CMS_VOS,list("out_buf","behar5_test")),"mover_max","1000"),
                 nlist("path","/storage/3","pgroup",merge(CMS_VOS,list("out_buf","behar5_test")),"mover_max","1000")),
    ## behar4 is the only box supporting all vos (for migration)
    "behar4",list(nlist("path","/storage/1","pgroup",merge(VOS,list("out_buf","behar4_test")),"mover_max","1000"),
                 nlist("path","/storage/2","pgroup",merge(VOS,list("out_buf","behar4_test")),"mover_max","1000"),
                 nlist("path","/storage/3","pgroup",merge(VOS,list("out_buf","behar4_test")),"mover_max","1000")),
);

#"/software/components/dcache/pool/pools"= {
#   tbl = self;
#   list = WN_DACHE_POOLS;
#   ok = first(list,k,v);
#   while (ok) {
#       m = matches(v,"([^\\.]+)(\\.(.*))?");
#       tbl[m[1]] = list(nlist("path","/scratch/1","size","125","pgroup",list("wn_pool"),"mover_max","1000"));
#       ok = next(list,k,v);
#  };
#  return(tbl);
#};
```

Poolgroup for grid publishing

Poolgroup linked to network unit

Poolgroup (one per box) linked to storage unit

Static list of nodes, could be a function

dCache installation and Quattor: ncm-dcache example code

```
## pools in pgroups
$real_exec="cd PoolManager\n";
my $tmp_exec="";
foreach my $pgrp (sort keys(%pgroups)) {
    if (1 == $pgroups{$pgrp}) {
        pd("$func: pgroup $pgrp is new, not checking it's current config.","i","10");
    } else {
        $tmp_exec .= "psu ls pgroup $pgrp -a\n";
    }
}
$real_exec .= $tmp_exec."..\nlogoff\n";
($exitcode,$output) = run_as_admin($real_exec,"true");

my @out=split(/\(PoolManager\).*\n/,$output);
## remove first and last from this split (this is the cd PoolManager and the ..\nlogoff part)
shift(@out);
pop(@out);

my $n=0;
while ($n < scalar(@out)) {
    ## the very first word is the name of the pgroup
    if ($out[$n] =~ m/^\W*(\w\S*)\s/) {
        my $pgrp = $1;
        if ($out[$n] =~ m/\spoolList.*\n((.*\n)*).*$/) {
            my @pppool=split(/\n/,$1);
            foreach (@pppool) {
                ## $_ now looks like eg:    behar_1  (enabled=true;active=26;links=0;pgroups=6)
                ## the output might contain some special characters (^M)
                ## \w\S* is to capture names with - in it etc. they just have to start with a \w
                if (m/(\w\S*)\s/){
                    if (exists($pgr_pool{$pgrp}{$1}) && 0 <= $pgr_pool{$pgrp}{$1}) {
                        $pgr_pool{$pgrp}{$1}=0;
                    } else {
                        $pgr_pool{$pgrp}{$1}=-1;
                    }
```

ssh -F config_file admin@host < /tmp/commands

dCache installation and Quattor: ncm-dcache example output

```
2007/01/12-15:31:07 [INFO] run_as_admin: function called with arg: cd PoolManager
psu ls pgroup beapps -a
psu ls pgroup becms -a

           ...

2007/01/12-15:31:09 [INFO] sys2:exec: ssh -F /root/.ssh/dcache_admin_config admin@localhost </tmp/
dcache_admin
2007/01/12-15:31:09 [INFO] sys2:output: Pseudo-terminal will not be allocated because stdin is not
 a terminal.

^M    dCache Admin (VII) (user=admin)


^M(local) admin > cd PoolManager
^M(PoolManager) admin > psu ls pgroup beapps -a
^Mbeapps
^M linkList :
^M poolList :
^M   behar4_2  (enabled=true;active=11;rdOnly=false;links=0;pgroups=9)
^M   behar4_3  (enabled=true;active=23;rdOnly=false;links=0;pgroups=9)
^M   behar4_1  (enabled=true;active=11;rdOnly=false;links=0;pgroups=9)
^M(PoolManager) admin > psu ls pgroup becms -a

           ...

2007/01/12-15:31:09 [INFO] config_pgroups:   pgroup beapps: 0
2007/01/12-15:31:09 [INFO] config_pgroups:     pool behar4_3: 0
2007/01/12-15:31:09 [INFO] config_pgroups:     pool behar4_1: 0
2007/01/12-15:31:09 [INFO] config_pgroups:     pool behar4_2: 0
```

Poolgroup exists

Pool already in poolgroup

dCache installation and Quattor: ncm-dcache future

- Add 170 features
- Maintenance problem has to be solved:
  - If dCache team accepts it, they could help with component
  - Best way: dCache team releases more complete dCache configuration tools
    - Easy to call them from within ncm-dcache
    - Easy to generate special configuration files (if needed)
    - Some of the code could even be reused/ported