Quattor : Managing a Complex Grid Site

Michel Jouvin
LAL/Orsay
jouvin@lal.in2p3.fr

CHEP07, Victoria
September 4, 2007

Agenda

- Quattor goals
- Quattor at a glance
- QWG Templates
- Challenge of site complexity : how Quattor helps ?
- Conclusion

Why Quattor ?

- Developped by European Data Grid WP4 (2001-04)
  - Efficient management of large fabrics with different type of nodes
  - Independent of grid middleware
- Management tool for reproducible configurations
  - Installation AND reconfiguration based on a common configuration description
    - Existing tools doing one or the other : Kickstart, APT...
  - Easy configuration cloning and modification rollback
    - Relaying a configuration must produce the same result
  - Configuration validation before deployment
- Service oriented
  - Easy mix of services on a machine
    - As long as the mix is supported (port conflict...)
    - No side effect of one service configuration on another service
  - Configuration consistency of common low level service

- Machine configuration described in term of final sta
  - No description of how to reach the state (actions)
- Abstract, service oriented description of machine configuration
  - Written in PAN language and compiled
    - PAN is a declarative-like language
  - Organized in “templates” that can be heavily factorized
  - Configuration consistency checked at compile time
    - Information schema and validation functions
- Configuration stored in a versioned central databas
  - Allow very easy rollback of any change
- Configuration implemented through “components”
  - Components are plugins running on client : 100+ availab
  - Extensible framework : easy to add new components
- Manage both installation and configuration changes

Why QWG templates

- EGEE/LCG MW installation tool : YAIM
  - Shell scripts configured through environment variables
  - Initial installation only, reconfiguration difficult
    - Not adapted to the need to frequently deploy patches
  - Machine oriented rather than service oriented
    - Assume a machine dedicated to a service
- QWG Templates : service oriented configuration provided as a set of standard PAN templates
  - Take advantage of NCM components being service orient
  - PAN templates are easy to organize by service
  - Easy mix of service on a machine
    - As long as MW supports the mix (port conflict...)
    - No side effect of one service configuration on another service
  - Configuration consistency of common low level service
    - Accounts, cron, NFS mounts...

LCG Quattor Working Group

- Official working group of LCG GDB
  - Mandate to ease and coordinate use of Quattor in grid context
  - Not a task force with significant manpower... Never been more than spare time from a few people !
  - Chairman : Cal Loomis
- Produce and maintain PAN templates and NCM components for EGEE/LCG MW configuration and deployment
  - First release for LCG 2.3 (2 years ago)
- Contribution to maintenance and development of Quattor
  - SCDB (Subversion based CDB) in replacement of origina (CVS) CDB

- Automatic generation of as many templates as possible
  - Per service RPM lists from MW descriptions
    - Unfortunately official list often incorrect
  - Done for the OS templates too (from RH description)
- No modification to standard templates for site customizations
  - Site configuration kept in 1 template
    - Based on variable definitions used by other templates
  - When sensible, definition of default values for MW parameters
    - Default value is used only if there is no explicit definition
- Take advantage of ‘Russian puppets’ feature
  - A separate template for each low level service (e.g. NFS)
  - This template included by others requiring it

- For each high level service, 1 template defining the whole configuration
  - Independently of the OS configuration
  - Sort of “public API” to templates
- 1 template for each major machine types
  - Just 1 line to put in a real machine profile
  - Includes relevant OS configuration
  - Generally several machine types can be combined on 1 machine
    - Restrictions come from MW

Templates Packaging

- 2 main sets of templates : OS and gLite
  - OS : 99% generated from RH/SL comps.xml
- Explicit release cycle with a publishedroadmap
  - gLite updates delivered as part of new template releases
    - First delivered a week after official release in pre-releases
  - Ability to select gLite update to install on a per node bas and independently of template releases
  - https://trac.lal.in2p3.fr/LCGQWG/roadmap
- Backward compatibility is an (personal) obsession
  - Installing a new release of templates produces the same configuration if new features are not used
  - A new release never requires a change to site configurat if new features are not used
- Set of working examples

Templates Benefits

- Added value by QWG templates compared to YAIM
  - SL4 64-bit support on WNs since spring 2006 (LCG 2.7.0
  - NFS configuration for shared VO areas or home dirs
    - Support many conf scheme, including dedicated NFS server
  - Very simple VO configuration : VO name added to a list..
  - BDII off CE since the beginning
  - Torque/MAUI v2 for several years
    - Including configuration examples for MPI/Short Deadline Jobs support
  - Torque submit filter customization
  - Gsissh accessible public UI
  - Non dedicated UIs for deployment on desktops or internal servers
  - Ability to close/drain a CE or selected WNs
  - Ability to restart all or selected WN LRMS clients

Profiles Examples

- All profiles for a specific machine types identical

```
object template profile_grid23;

include machine_types/lal_wn;

include repository_common;
```

- WN type defined using generic templates + variable to set site specific parameters

```
unique template machine_types/lal_wn;

# GRIF specific configuration for a WN
variable WN_CONFIG_SITE = "pro_wn_grif";

include machine-types/wn;
```

- IP address, OS version... specified by variables either in the machine profile or shared

```
variable OS_VERSION = nlist(
  escape("grid11.lal.in2p3.fr"),"sl307-i386",
  escape("grid15.lal.in2p3.fr"),"sl440-x86_64",
  escape("grid16.lal.in2p3.fr"),"sl440-x86_64",
  escape("grid17.lal.in2p3.fr"),"sl440-x86_64",
  escape("grid23.lal.in2p3.fr"),"sl440-x86_64",
);
```

Template Example

- Machine HW configuration is defined in a site specific template using generic templates

```
structure template hardware/machine/200/48/ibm1/slot30;

"location" = "200_48_ibm1_30";
"serialnumber" = "KKWMZ7T";

"cpu" = list(create("hardware/cpu/opteron_248"),
             create("hardware/cpu/opteron_248"),
             create("hardware/cpu/opteron_248"),
             create("hardware/cpu/opteron_248"));

"harddisks" = nlist("sda", create("pro_hardware_harddisk_sata", "capacity", 80*GB));
"ram" = list(create("hardware/ram/generic", "size", 8192*MB));

"cards/nic" = nlist("eth0",create("hardware/nic/tg3"),
                    "eth1",create("hardware/nic/tg3"));
"cards/nic/eth0/hwaddr" = "00:11:25:C4:20:DA";
"cards/nic/eth1/hwaddr" = "00:11:25:C4:20:DB";
"cards/nic/eth1/boot" = true;
```

- All MW variables grouped in one file

```
# SITE DEFINITIONS ----------------------------------------
----------------
# ----------------------------------------------------------
----------------

variable SITE_EMAIL ?= "grid.support@grif.fr";
variable SITE_NAME ?= "GRIF";
variable SITE_LOC ?= "Orsay, France";
variable SITE_LAT ?= 48.699262;
variable SITE_LONG ?= 2.170686;
variable SITE_WEB?="http://grif.fr/";
variable SITE_OTHER_INFO ?= list("TIER 2", "IN2P3-CC");

variable SITE_DOMAIN ?= "lal.in2p3.fr";

variable INSTALL_DATE ?= "20060101120000Z";


# SERVICE LOCATIONS ----------------------------------------------------------
# ------------------------------------------------------------------------

variable CE_HOST ?= "grid10."+SITE_DOMAIN;

variable SE_HOSTS ?=
  list("grid11."+SITE_DOMAIN,
       "grid05."+SITE_DOMAIN);

variable SE_TYPES ?= nlist(SE_HOSTS[0], "disk",
                           SE_HOSTS[1], "SE_dpm");

variable SE_ACCESS ?= nlist(SE_HOSTS[0], '/some/value',
                            SE_HOSTS[1], '/some/other/value');
```

```
variable SE_ARCH ?= nlist(SE_HOSTS[0], 'multidisk',
                          SE_HOSTS[1], 'multidisk');

variable SE_HOST_DEFAULT ?= SE_HOSTS[0];
variable SE_HOST_DEFAULT_SC3 ?= SE_HOSTS[1];

variable SE_DPM_DISK_HOSTS ?= nlist("grid07."+SITE_DOMAIN, "1");

variable LFC_HOST   ?= "grid07."+SITE_DOMAIN;

variable RB_HOST   ?= "grid09."+SITE_DOMAIN;
variable PX_HOST   ?= "grid02."+SITE_DOMAIN;
variable BDII_HOST ?= "grid01."+SITE_DOMAIN;

variable MON_HOST  ?= "grid08."+SITE_DOMAIN;
variable GRIDICE_SERVER_HOST ?= MON_HOST;

variable GRIS_PORT ?= 2135;


# BDII CONFIGURATION ----------------------------------------------------
# -----------------------------------------------------------------------

variable BDII_URLS ?= {
  urls = nlist(
     "CE", "ldap://"+CE_HOST+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o=grid",
     "LFC", "ldap://"+LFC_HOST+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o=grid",
     "LFCLAL", "ldap://"+'grid14.lal.in2p3.fr'+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o
     "RB", "ldap://"+RB_HOST+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o=grid",
     "PX", "ldap://"+PX_HOST+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o=grid",
     "MON", "ldap://"+MON_HOST+":"+to_string(GRIS_PORT)+"/mds-vo-name=local,o=grid");
```

Challenge of site complexity

- Many ways a site can be “complex”…
  - Different types of machines : grid, non grid, desktops, DB servers…
  - Distributed/federated sites are a growing reality
    - In particular in LCG
- … Probably every site is “complex” !
- Management challenge : factorize the management still preserving the necessary autonomy of every part
  - Many common part in configuration/administration of different classes of systems…
  - … But a monolithic solution doesn’t match the need
  - Autonomy is important for high availability and dissemination of expertise…
  - … but duplication is a huge waste of manpower
    - Sharing recipes is not enough

Grid Technical Challenges

- Build one homogeneous grid site from 5 sites
  - Have a predictable behavior of the whole site
  - Easy and transparent relocation of resources and load
  - Same quality of service whatever is the number of administrators at each site
- Leverage administration of the whole site
  - No possibility to hire anybody for the project : required t build a technical team with existing people
  - Very few dedicated grid person
  - Take the most from existing manpower : avoid duplicatio
- Increase the overall expertise
  - No site large enough to take control of the others
  - Very few grid “experts” at the beginning : 2-3 people
- Integrate operation at each site with non grid syste
  - Generally the same people managing both
    - DAPNIA is an excention

- Being 1 site created a driving force for funding mutualization between various groups
  - Result : 1 year in advance on original planning
- Expertise dissemination is a reality
  - Between 1 and 3 grid admin “experts” per site
- GRIF-wide technical team is a reality
  - Lots of everyday cross-site interactions
  - 1 face-to-face meeting per month
  - Configuring a service, installing an update… can be handled by one person for the whole GRIF
    - Can change from one time to another
- Each site still has control of its own resources
  - Local customization possible : storage integration, authentication infrastructure, network params…
  - Integration of operations with non grid systems underway at most of the sites.

Multiple clusters and sites

- Standard templates and site specific templates are in separate hierarchies
  - 2 level of site specific templates : clusters and sites
    - A site is not necessarily a geographical concept
    - A cluster is an arbitrary grouping of machines
    - A cluster can belong to several “sites”
  - Site customizations done through PAN variables took as input by standard templates
  - A template can be put/redefined in any hierarchy
- GRIF configuration DB contains all grid systems from all GRIF sites + all LAL non grid systems
  - 1 site ‘grif’ containing all parameters common to all GRIF sites for grid systems (GRIF is only grid)
  - 1 site per GRIF site defining parameters specific to a site
    - E.g. : network parameters, site admins…
  - Grid clusters belong to site ‘grif’ + local site
  - LAL non grid systems belong only to site ‘lal’

Controlling access rights

- An inappropriate modification to standard/common templates can potentially break a lot of systems…
  - Quattor will never apply RPM changes that break dependencies for installed or new RPMs
  - Quattor has the ability to rollback any change as long as the machine is reachable and the Quattor client installed
    - Includes ability to downgrade RPMs
- SCDB allows to restrict right to modify some parts o the configuration
  - Use SVN ACLs : based on the userid used to commit changes
  - Can be done at any level of template hierarchies
  - GRIF : a limited number of persons can modify standard site ‘grif’ templates, everyone has full control of its site
- Without mod rights to standard templates, still possible to redefine a standard template in site/clus
  - Allows everyone to contribute mods by preparing them i

Usage of Quattor

- Quattor usage increased over last years
  - 40+ sites, mainly in Europe
    - Several group of sites managed from a unique database
  - Some countries “quattorized”
    - France, Spain, Ireland, Belgium
  - Still very few usage outside LCG : Ireland
    - Quattor sites are not dedicated to HEP
- Wide range of site size
  - CERN : ~5000 nodes
  - Several T1s : NIKHEF, CNAF, PIC
  - Many T2s and T3s, including large ones (GRIF, DESY) and small ones (<50 nodes)
- Main usage is management of grid resources...
  - Starting point to use Quattor
- But growing non grid usage at these sites
  - Site internal servers, desktops, virtual machines (XEN)

- Usage increasing : some sites with 200+ nodes
  - GRIF : 5 sites, 200 machines, grid + non grid usage, 1 S
    - +250 machines arriving end of September
  - 6 french T2/T3s : 1 SCDB per site, 10-50 nodes/site
  - BEGRID : 3 prod sites, 200 nodes, 1 SCDB (+ 1 testing s
  - Irish grid : 15+ sites, 350 nodes, 1 CVS SCDB
  - UAM : 700 nodes, 3 clusters, 3 SCDB
  - Valencia : QWG, SCDB, 175 nodes
  - Aachen : 1 SCDB, 35 WNs, 7 dCache SEs, 130 UIs
  - Protvino (Moscow region) : still LCG 2.7.0...
  - PIC (T1) : for WN currently
- Other large sites actively considering switching
  - DESY

- Quattor is a unique tool to leverage administration o medium/large sites
  - GRIF showed efficiency of sharing management tasks
  - To be acceptable, each site needs to be convinced that it win something and not loose its autonomy
    - Main wins are ability to start quickly and to focus on services witho being overloaded by the basic operations
  - To build a distributed technical, a common management framework allowing to factorize management preserving autonomy is critical
    - Quattor allows much more than sharing expertise and recipes
- QWG effort is critical to leverage Quattor value
  - Allows out-of-the-box usage of Quattor for grid
  - Hides complexity of setting up and maintaining a site
- Quattor is now a community-backed product
  - Main developers and expertise outside CERN
  - Remains a critical product for CERN ; never been a probl

- QWG site : http://trac.lal.in2p3.fr/LCGQWG
  - Main documentation : pretty complete
  - Description of QWG templates (OS and MW)
  - Description of SCDB
  - Description of PAN language
  - Actively updated
- Quattor main site : http://quattor.org
  - Mainly basic architecture and NCM components
  - Less actively updated…
- Other presentations :
  - Hepix DESY 07 : https://indico.desy.de/contributionDisplay.py?contribId=18&sessionId=46&confId=257