Quattor Workshop – LAL – 22-24/3/06

http://agenda.lal.in2p3.fr/fullAgenda.php?ida=a0627

Quattor Usage Survey – G. Cancio

Huge increase of Quattor usage in one year. Now 39 sites with 18 CDB instances (3 testing)

- More than 6500 nodes managed by Quattor
- 2 sites outsite Europe

CDB instances size :

- 2 sites > 1000 nodes
- 7 sites ~100 nodes
- 9 sites ~10 nodes

Mainly running SL3/i386. Interest growing in SL4 and x86_64

Subsystem usage :

- AII : all instances but CERN
- SPMA : all instances (but some using YUM too)
- SWrep : 13
- CDB : 11, SCDB : 5

LCG configurations :

- QWG only : 6
- YAIM only : 3
- Combination : 5

Lemon : 5 in prod + 3 testing

Common local extensions :

- Web interfaces
- Local integration scripts
- Local NCM components (in particular at CERN)

Most common problems / improvement requests :

- Namespace for CDB
- Panc speedup
- LCG template complexity

- Template/profile browsing tools

Site Reports

BEgrid – Stijnde Weirdt

BEgrid : 5 sites managed independently, mainly with Quattor

- Most people doing the same thing

Proposed solution :

- One central server based on SCDB + certs
- SWrep for RPMs repositories
- Use of http cache

Don't plan for central administration. Administrators produced tagged version. Site administrators choose to deploy them or not.

Problem of sensible data : no standard encryption in pan, thus using dummy values in the central config and updating it on deployment at local sites.

Currently one site running the configuration and another one joined last week.

Success depends on people willing to contribute and the ability to keep template structure clean

- Is it really easier ? Certainly for newcomers…

CERN – G. Cancio

Quattor used to manage > 3800 nodes (not including desktops)

- Special justification needed for not having a system quattor managed. Resistance decreasing

CDB/CDBSOAP

- Slithly differenc core template set (older than public release)
- YAIM for LCG install
- Desktops have a local, common XML profile (not in central CDB)
- SPMA/SWrep except for desktop where YUM is used for greater user flexibility
- CDB2SQL backend heavily used

Problems encountered :

- Need for higher level workflow apps able to interact with CDB (LEAF/HMS) and connection with ticket system, Web interfaces to display CDB summaries
- CERN-CC relies on CDB as a central database for HW description. CDB2SQL doesn't fulfil all requirements (how to store information about a non machine element, like a rack).
- Deployment of scurity updates is painful with explicit package lists. Developped a script to generate package list out of a YUM repository. But never tested outside CERN and doesn't do anything about dependencies management (need to be done manually)

Major developments :

- ACL support for templates
- Support for namespaces
- Soap layer improvement for better perfs and improvement of concurrent usage (can reach 30)
- CDB backend : rewrite in progress in multithreaded python
- SOAP version of SWrep (using same auth as in CDB). Collaboration with BARC (India).
- Rpmt-py in replacement of rpmt : more fault tolerant, better integration with Yum.
- CERN manpower limited : 1.4 FTE (including 1 from INFN)

Quattor release cycle : need to provide an easy to use tool to build a new release with all components.

- 1.2 planned as soon as CDB namespaces are ready

PAN compiler :

- Better control of what can be redefined in a template
- Easier programmatic IF by adding a new 'data' type template
- Direct back end interfacing lile XML DB or Oracle

Linux for Control Machines at CERN – M. Shroder

Very specific issues :

- Limited network connectivity
- Don't want a 'deploy everything' approach
- Fine control on who can modify what
- Diskless system : spma and ncm cannot run on clients. Must be run on the boot server.

Lack of dependency management by the compiler is an issue as you detect problems when you run SPMA.

Need for a GUI to navigate templates.

Need a tools to sync several CDBs instances : separate instances required to avoid interference between different clusters.

Need a tool to easily check what version is used for a given product.

CNAF – A. Chierici

Quattor used for everything (LCG and non LCG system)

- Use AII, SPMA, CDB but not SWrep
- ~1000 nodes, only SLC3. Looking at SLC4
- Still some concerns about using Quattor to manage specific servers (mail servers, castor servers…)
- More documentation needed to help storage specialist (or other services) to better understand how they can use Quattor with the flexibility required (installation of specific drivers..).

Thinking about using Quattor to manager T2s in Italy.

Used QWG in 2.6. Will probably move to YAIM for 2.7

- Need to install ig-lcg, Italian customization of LCG
- Need to have direct support and suggestions by CERN experts for solving tricky issues.
- QWG template maintenance too dependent on Cal availability

AII : implemented a –rescue option to run a rescue environment on failing nodes

- Runs as a Linux-in-RAM, allow for memcheck and disk formatting
- Use a specific PXE config file, rescue.cfg

Requests for the future :

- A well defined roadmap
- Better integration with tools like Lemon
- More components : filecopy is the most used with pros and cons… (e.g. ssh component not managing ssh_config)

DESY – U. Ensslin

Quattor used differently on WNs and other nodes :

- WNs fully installed with Quattor + QWG templates
- Other nodes : Quattor for OS install only, YAIM for LCG services
- Currently ~150 nodes managed by Quattor (Hamburg only)

1 SWrep + several "Quattor instances" (CDB/AII) used for testing

CDB CVS stored in AFS to ease coordination between developers.

- Template naming scheme modified to allow for DESY specific version of templates.
- **Developped a tool to display pan template include hierarchy (lth). Doesn't handle templates created by create() function.**

Developped a Web based interface to install machine without having to know about templates and support bulk installation

Some template/component modifications :

- executecmd : allow to execute a command on a node in a pre configured list
- ncm-afsclt and ncm-sshd improved
- Not yet committed to repository…

Requests for the future :

- Erratas management
- Panc compile time
- QWG templates too complex. MW people looking for simpler solutions : YAIM is ok for them. Need some help on how to use ncm-yaim.
- Partitioning issues
- Quattor for Solaris

Template management remains difficult :

- Many sources of templates
- Many releases for MW, OS… more and more template version. How not to get lost ?

GridIreland – S. Childs

18 sites spread around Ireland managed centrally.

- ~190 nodes

Quattor configuration

- No CDB : CVS + ant
- No SWrep : resync http servers, substitution of server names at local sites

- Wide usage of VMs (Xen) : ~95% of service nodes
- Hierarchichal config : EGEE->GridIreland->Site
- Implemented ability to compile one specific site

Minor issues :

- Compilation speed
- Need components to manage VMs
- Sometimes cumbersome command syntax
- Improvement needed in the profile modification notification process

NIKHEF – R. Starink

NIKHEF + SARA are T1s. Also support for national projects.

- ~150 nodes in production, ~15 for testbed
- Non standard environment : CentOS

Quattor configuration :

- No CDB : CVS + ant
- AII, NCM, SPMA

LCG2 :

- Installation done with SPMA
- Configuration : migrating from QWG templates (+ local modifications) to ncm-yaim (+ local modifications)

Successes : SW version management + automated installation and configuration. Works well.

Problems :

- Upgrading usually full of surprises
- LCG2 templates : complex structure, many local modifications to standard profiles, backward compatibility, insufficient ChangeLogs…
- Tools : lack of management of dependencies, profile comparaison

Quattor for managing large mail systems – J. Gato

A large mail system characterized by a large number of machine running a lot of different services (SMTP, LDAP, Db, Web…)

Andago involved in deploying and maintaining such systems for customers. Many duplication of effort, problem to maintain consistency. Started TOMAS (Quattor+Nagios+FailTolerance+OSCng..)

- Quattor to be used for package deployment and configuration
- Interested by Quattor supporting other distritution (and alternative packagers)
- Open source (exact license not yet decided)

Andago plans to develop components for several services.

- Many have to be written from scratch (http, anti-spam, MySQL…)

1st release of TOMAS planned 31st of May. http://tomas.andago.com (jgato@andago.com)

CDB / SCDB Status and Directions

CDB Evolutions – M.E. Poleggi

CDB : 3 tiers architecture

- SOAP client :cdbop interactive/batch shell
- SOAP middleware : Apache + cdbsoap CGI
- CDB backend : CVS

Template flat space : class distinction only possible through template names, complex administration with several clusters, no easy way to limit the scope of user actions.

- Namespace should allow to solve this

Authentication problem : weak security due to ciphered password

- Pwd stored in CDB server filesystem
- Non interactive clients need to store clear text pwd
- Possible solutions : certs and Krb. Certs almost done (cdbop), beta testing to start : checks only the client cert (do we need server cert check ?). Currently no support for grid proxy (theoretically possible).

Authorization issues : only 2 classes of templates (pro + others). Authorization currently based on class only

- Solution : namespace + groups + ACLs

Performance issue : one network connection for each operation. Plan to send, process and return a list of items over the same network connection.

- Almost done, huge improvement for bulk modifications

Namespaces : like a directory.

- Allow homonymous templates
- Easy to enforce access privileges per namespace
- Implemented by pan loadpath
- Mostly done, still in progress for cake

Privilege enforcement through ACLs : semantic similar to Unix FS

- <item>:<user/group>=<right>
- Support for groups
- Based on 2 plain text files on CDB server : editable with cdbop or a text editor

Future directions :

- Web service extensions : unify parsing/packing of I/O data, provide a WSDL description
- Fine grained CDB locking with fair queuing
- Parallel compilation of independent templates : how to extract dependency information in advance ?
- Use mod_perl instead of CGI : could improve a lot response time at a price of lack of portability

SCDB – C. Loomis

Design goals :

- Multi-cluster mgmt with a single database
- Hierarchical arrangement of templates
- Treat configuration directly as code management : cdbop hides a lot of version management features
- Usable offline
- Must work in any environment and be secure

Implementation :

- Relies as much as possible on existing tools
- Subversion : atomic commits, directory management
- Ant : equivalent of make in Java, framework for including new tasks, method for executing simple workflows (task dependencies)
- Eclipse (optional) : provides GUI interface, integrates with SVN and ant, syntax coloring for pan

Dependencies :

- Apache 2 server
- Subversion server/client

- Java 1.5
- Pan compiler
- Ant
- Optional : Eclipse with Subclipse, javasvn, colorer editor, SunShade plugins

Status :

- Latest release : https:/svn.lal.in2p3.fr/LCG/QWG/SCDB
- Components : build files (300 LOC), quattor.jar (5 ant taks, 2k LOC), SNV hook script (200 LOC). Easy to maintain.
- Ant task : compilation (PanSyntaxTask, PanCompileTask), deployment (SvnTagTask, NotifyClientTask), RPM repository management (RepositoryTask)

Major changes planned : none, works very well

- Would like to remove need to log onto the server for AII tasks

Minor changes :

- Reorganization of templates
- Integration of new pan compiler
- New tasks : OS updates, patches, repository template upgrades
- Integration of certs, VOMS

Discussion

G. Cancio : CERN needs SOAP interface because some the tools used (Remedy…) can talk only SOAP.

AII Overview and Future – R. Starink

New maintainers since May 2005 : Cesar Lobo (UAM) and Ronald Starink (NIKHEF)

Current stable release : 1.0.32 (August 2005)

- Use new NCM template processor for setting partitions
- Support for https
- A few bug fixes

Feb. 2006 : Jorge Izquierdo joinded, continuation of work C. Lobo on partitioning

March 2006 : release 1.0.36 by M .Jouvin

- LVM support

- Improved package resolution
- Replace rpmt by rpmt-py

6 open bugs :

- rpmt requires multiple invocation
- aii-shellfe should not have log files
- RPM upgrade should be optional : probably related to the times where RPM downgrade was needed. No longer the case.
- Extending device naming schemas : more a pan template issue
- Assumption on the profile name

Taks and enhancements :

- ncm-components for managing PXE/KS files
- ad-hoc scripts in KS pre/post install
- LVM support : done, one bug in template in 1.0.36
- --rescue option : done
- Fetch certificates using sindes

Future plans :

- Fix bugs
- Compliance with SL4 and x86_64
- Improve documentation : complete description of all options
- Review NCM template processor : convenient but not easy to maintain. Cal : XSLT could be an alternative option to translate XML to something else.

Lemon – G. Cancio

Lemon :

- Monitoring agent running on each node sending data to the server. Information can be requested by the server with hearbeat between server and sensors or sensors can work in pure push mode.
- Sensors to measure various metrics. Each sensor provides metric classes (parse log file, list process matching a criteria…)
- Server : flat file or Oracle based
- Display framework based on RRD/Web

CERN Web pages at URL https://lemonweb.cern.ch/lemon-status (requires a NICE account).

Can get perf data and other metrics like uptime, kernel version used, reboot time…

Can link to CDB / XML profile of specific machine

May define clusters and subclusters

- Cluster definition can come from CDB
- May aggregate resources by VOs (using group information)

Can do metric comparaison between node and/or clusters

Many sensors available, including database sensors (Oracle).

Raw monitoring data are never aged out automatically. Must be requested explicitly.

- Automatic table compression on Oracle

May launch recovery actions on certain conditions

- Implemented through an alarm sensor using exception definitions (exceptions are basically a kind of metric)
- Can set a limit to the number of retry for recovery actions
- Correlation engine (CMDaemon) but not currently used at CERN

Working on an associated alarm management tool (GUI). Will implement alarm reduction

- Horizontal : if several nodes report an alarm, report it once as a 'cluster' problem
- vertical : mask some alarms in presence of others
- Will use PySQL

Quattor-Lemon integration :

- CDB can hold definition of all sensors, metric classes and instances, exception handling…
- 1 ncm component to manage lemon client
- 1 ncm component to manage Oracle based server (no component for flat file server)
- All the Lemon information under /system/monitoring
- Predefined templates available in Quattor CVS but out of date. Need to get those from tarball

Future developments :

- Secure communication between Lemon components. Based on SSL. Done, under stress test. Both TCP and UDP.
- Alarm system
- Service Level Status System to give a user oriented view of services with a view of how appropriate is the service level (fully available, affected, degraded, non operational…). Under development.

- Planning a “proxy” sensor for machines that cannot be monitored directly (e.g. Windows). The proxy sensor will return metrics based on some tests with the target machine.

Sensors are common with gridIce but the config files are not (yet compatible). GridIce people need to upgrade to the new configuration scheme (directory based) : work in progress

- There can be issues running both GridIce and Lemon on the same machine. Basically need to run 2 separate agents until the new GridIce version is available.

HW required for the server : don’t need a very fast machine. No real scalability problem.

Oracle backend can be used on Oracle Express version (free). But :

- Oracle Express has no support for backup and partitioning (critical for performances)…
- Oracle doesn’t commit that Oracle Express will exist in the future…

PAN Status and Developments - Cal

Current release : 4.0.2

- Full profile compression added (as an option) and thus removed element compression
- Complete load path is combination of command line values and the ‘loadpath’ global variable in templates (specify relative directory).
- Runs on Linux (rpm), Solaris (pkg), Mac OSX (dmg/pkg), Windows (no packaging, source only, must be built with Cygwin)
- Pan specification up to date with current implementation
- Still no user guide…

Annoyances :

- argv cannot be used as a function argument (not implemented as a list)
- Iterators : not possible for global variable, cannot have multiple iterators on the same variable
- Variable masking (silently) by first() and next()
- Global/local variable collisions : cannot modifiy global variable from a function and cannot use the same name for global and local variables. Main workaround is name convention for variables (e.g. global variable names in uppercase).
- Path specification : DML needed when a path element requires encoding (‘/alpha’ = {r=self;r[escape(‘a/b)’] = 2; return(r);};). Possible solutions through syntax extensions, e.g. loosening restrictions on path elements with a syntax like ‘/alpha[a/b]/gamma’ = …

Performances issues :

- Memory usage : no garbage collection at profile level
- Reference counting for temporary objects to implement some basic garbage collection for temporary object. Not efficient at all
- Hand tuned string/memory management : very Linux 32-bit specific. Noticeable perf problem on 64-bit
- Single threaded : prevents multi CPU usage.
- Frequent (and expensive) object copies : any reference to tree and global variable results in a copy. Keeping derivation information (where the property/resource has been set) makes this worse.

Possible solutions :

- Memory : use standard memory management, use garbage collection library
- Multi-threaded implementation is really needed but requires a deep re-engineering of the code
- Immutable properties : no need to copy them if we remove derivation information

Proposal :

- Rewrite in Java : portable between platform, advanced garbage collection, easy integration of multi-threading, logging and regexp part of the language spec, serialization for object files, completed templates is trivial and should allow perf improvement (save intermediate object for reuse).
- Speed issues : probably slower for a small number of profiles (jvm startup overhead) but speed up for dual CPU (expect 2x) and speed up from streamlined implementation
- Status : Complete JavaCC pan grammar exists
- Skeleton implementation but no complete prototype available yet

Proposed language changes :

- 'define' keyword complicates grammar in many places. Optional currently, removed in a future release to simplify grammar and speed up compilation
- 'delete' keyword : conflicts with delete(), propose replacing with 'null' value that could be assigned to a path ('/alpha' = null).
- Automatic variables upper-cased (SELF, OBJECT, LOADPATH, ARGC, ARGV) : more consistent with convention, suppress conflict with 'object' keyword, migration period with both supported

Functions : requests for new features

- Character handling functions
- Substitute() to allow variable replacement in strings
- Push/npush : non obvious behaviour (need to return self). One possibility would be to add a 1st argument telling the push destination, could allow merge of push and npush.. Need to be moved inside the compiler for efficiency

- merge() : error thrown for duplicate nlist keys. Can allow merge of duplicate keys if the value are identical or assure that there is no duplicate keys in the final list? Is there any unique solution or do we need a configurable behaviour. Could be moved to compiler for efficiency.
- Bitwise operation support

Other requests for enhancements :

- Data template : will only allow assignmentsto compile-time constants
- Single inclusion : allow a template to be included at most one time (similar behaviour as for declaration templates but for other types).
- Variable includes : implementing if then else will complicate a lot pan grammar and will break non procedural nature of pan. Possible to implement ‘include {dml}’.
- Authorization : by template (limit what values can be set in a template), by user (limit what values a user can set)
- Default values inside the record definition (declaration template)

Roadmap suggestion :

- Freeze current C/C++ implementation
- Work on Java re-implementation (6 months estimated)
- Test both implementation against each other and choose the most sensible option

Discussion

Basic agreement on evaluation of Java re-implementation in a 6 months timescale.

- Need to implement some of the most urgent RFE before freezing (include {dml}, single inclusion, default values in record definitions)

G. Cancio : CERN has began to turn several templates included in each node (like sw packages) into an object template included in the node template through value(“//external/template/path”). This has dramatic impact on performance (4x to 5x).

R. Garcia : would like to get a profile schema definition into XML (XMLSchema ?).

NCM Components – M. Jouvin

Notes missing… See presentation.

RPM less components :

- Mixed feelings…

- **If implemented, both RPM and RPM less format should be provided for each components (modification of quattor build tools)**

Component configuration outside /software/components

- Apart from convenience, main reason should be information that can be shared between several components
- No consensus about a network component configuration location : should it register directly for changes where the information is (generally /system/interfaces, e.g. ncm-network) or transparently collect the information from where it is to a configuration area for the component (ncm-ifconfig). Many pros and cons for both approaches. Using information directly from /system area requires everybody agrees on and implement a common schema : not currently true (e.g. CERN !).

QWG Templates - Cal

Current releases : LCG-2.7.0-4

Overall design goals

- Be as service oriented as possible
- Be reasonably flexible to accommodate local site configuration

QWG problems :

- Complex structure : cost of flexibility, Grid services not "service oriented" (lots of dependencies between services), VO specific configuration
- Backward compatibility : service themselves are often not backward compatible (e.g. complete rewrite of config information for Info System between 2.6 and 2.7), VO specific improvements, VO specific config
- Poorly documented (undocumented) : difficult to start, example not always working/appropriate

YAIM

- Set of scripts maintained by the GD team at CERN
- Quattor component exists, used by many (written/maintained by CERN) : basically does nothing except defining variables required by YAIM and kicking YAIM script.

YAIM problems : not Quattor specific…

- Lots of assumption on site config
- Not very flexible
- No effort for reproducible down/updates
- Configuration fixes difficult (generally requires redeployment of RPMs)

- Poor documentation (essentially undocumented)

gLite Configuration :

- Standard file format with complete examples
- Configuration scripts with a consistent interface
- Quattor interface : script to generate pan templates, 1 component for all gLite services
- Problem : scripts often interfere with other components (accounts mgt…), problem with reproducible down/upgrades (same as YAIM), configuration fixes difficult (same as YAIM)

gLite 3.0 : should be released soon on PPS (+ 1 ½ month to production)

- Unclear they will go on with current gLite configuration mechanism
- Could move to a mix of YAIM and current gLite

Common issues :

- Manpower to maintain QWG templates
- VO configuration
- Documentation

Discussion

VO configuration :

- 1st implementation : for each VO under /system/vo, a nlist with the simple name of the VO being the key. For each VO, describes the specific 'services' (myproxy, rb…). Very difficult to have this information entered in a flexible way.
- Current situation : added a bunch of function. Very fragile : doesn't work if there is not the right path on the left hand side.
- New approach in QWG 2.7.0 : one global database VO_AUTH containing all the info for all the VOs and a few functions (mkgridmap…) processing the VO_AUTH information to produce the required information (gridmap…). More flexible as VO_AUTH has no schema information : information can be structured in the way the more appropriate for its final use (component schema). If agreed, could completely remove /system/vo. VO_AUTH created from structure template (1/VO). This structure template could be generated by a script from a central database.
- Future : should move VO services location out of VO declaration template

gLite :

- Need to feed developer on what can make our life easier… or worst. Important if we can have a common standing from all Quattor sites.

LCG template maintenance :

- Cal : could probably move for one person having the responsibility for all components to several people in charge of different services, as services are more or less independent.
- S. Childs propose to participate to beta testing of QWG templates
- QWG template mainly used by GRIF, Ireland, VULB, DESY (partly). Should concentrate on producing documentation and polishing templates so that a site can import the template tree for a MW version and have it running without any customization other than a few site dependant profile.

Namespaces : For what ? Which ones ? (discussion)

Cal : some base namespaces could be : os, pan, lemon

- Not really useful for LCG templates (can be just one namespace)
- For OS, os/version-arch ?
- Namespace should reflect responsibilities for component maintenance

Components :

- Proposal : component/<component>/defaults (equivalent to current pro_software_component_<component>).

OS : os/pkg/<pkg-definition> (e.g. base.tpl)

Pan standard declarations/functions : pan/*

Lemon : one specific namespace , lemon/*

Hardware : hardware/<cpu,ram,nic,machine…>/

Middleware : grid/<lcg|glite>/<service|config|machine>

Quattor : quattor conventions (schema…) : quattor/

Remark : namespace relative to load path.

TOMAS3 – R. Garcia Leiva

http://tomas.andago.com

Andago : company targeted at providing solutions for Linux based on open source technologies.

TOMAS3 : Toward an Open Management Architecture for Systems, Software and Services

- An open architecture where you can plug your existing products
- A production qualitiy product allowing to manage your own resources
- 2 pilot customers : Eroski (supermarkets), Univ Autonoma de Madrid

TOMAS3 based is made of several components :

- Deployment and configuration management : Quattor
- Service Management
- HW management
- Monitoring based on Nagios as rather than Lemon (better alarm management currently). Lemon could be used too.
- Fault tolerance
- Integration for acceptance by enterprise (documentation…)
- Communication between each component through Web services allowing to replace any component
- Compliance with management standards

Roadmap : try to use short cycles to get early feedback (6-12 months)

- 1st release : end of May
- 2nd release : end of 2006. Cycle open to the community.
- Will try to follow the schedule even at the price of delaying some functionalities as feedback is crucial.

Using CIM model in Quattor : currently no clear model for information managed by Quattor.

- CIM is a standard model, implementation neutral, object oriented description of configuration
- Advantage : industry driven standard will help acceptance by enterprises, should help future integration with web services.
- Problems : very complex model, not everything so well defined, object oriented approach difficult to implement with pan, migration will require a rewrite of all components

German :

- looked at CIM schema during the early times of Quattor but appeared unnecessarily complex. At the end Quattor would use only 10% of the schema.
- Other open source solution like ROCKS, LCFG have also their own schema, in fact much more basic than Quattor schema.

Using WSDM (Web Service Distributed Management) in Quattor :

- Open standard supported by industry
- WSDM specify how to manage distributed resources (even resources out of the scope of Quattor : printers…)
- Both Management Using Web Services (MUWS) and Management of Web Services (MOWS)
- Based on a producer/consumer model, support for discovery through web service endpoint
- New devices to come with support of WSDM
- Integration of devices without native WSDM can be done by a WSDL proxy
- Problems : doesn't fit very well with Quattor architecture, MUSE (open source server) doesn't provide any client, no open source SOAP server with support for SW addressing.

Cal :

- Don't expect any grid machine to have a WSDM interface. WSDL proxy is the only viable solution
- Configuration changes require support of a basic transaction module to be able to process a set of changes as one operation to avoid "intermediate states".

Quattor Future Directions and Roadmap – Discussion

Agreement to have another meeting in 6 months (early October ?).

- 2 days over 3 days ?
- Could imagine a session with // working groups of interest (e.g. SCDB, Lemon…)
- Fee free meeting is a good approach (and let everybody take care of the dinner…)

A Quattor release should be only the core components, not the templates (even the core templates) : CDB, SCDB, AII, SWrep.

- Templates must be released separately and namespace issue must be handled in template, not core components : have a separate set of templates for standard templates (quattor/pan), core, lcg.
- Ncm-components : should provide some kind of package of production version for components
- Need one wiki page documentating what's the production version for every subset

What about a tutorial on Quattor during T2 workshop

- Need to agree on one Quattor implementation
- Need to focus on T2 issues, in particular distributed T2
- May be no a real tutorial, more demonstrations and use cases

CDB :

- CERN will complete current developments and release them to the community (namespaces, ACLs…). Timescale : ACLs in March, namespaces in April/May.

QWG templates : should tighten the links between the 3 sites using them

- Trac site created and will be advertised to the list
- Try to use these templates as a testbed for implementing namespaces as discussed. Need to guarantee that build tools are able to generate both a version with and without namespace during the transition period. Start with pan, quattor and os templates
- Try to implement German suggestion of using object templates for package list (at least on worker nodes) and may be VO information.
- Beta testing of service by GridIreland and VULB (in particular for glite).

Lemon templates :

- Last version to be provided next week
- Provide some feedback on integration of last configuration scheme for sensors into GridIce

AII :

- Have quickly a new release with main bug fixes (need to fix kickstart template) and addition of –rescue

PAN compiler :

- Implement variable includes, single includes and default values in declaration templates
- Freeze and reimplement in Java for comparaison and evaluation

NCM components :

- Identify CERN components that should be made generic and move them to core (e.g afsclt)
- Build an index of existing components from CVS with a script and update information on CVS (README files) if needed.
- How to build a release of components ? Who can be a release maintainer ?

Savanah :

- Our main bug tracking, want to disable task list
- Workflow should be clarified : try to do it in coordination with other group. Ongoing discusson at CERN. Conclusion will be sent to the mailing list.

- No consensus on entering something into Savanah for every change (at least tag) to CVS.