Automated management of large fabrics with ELFms

Germán Cancio for CERN IT/FIO
LCG-Asia Workshop
Taipei, 26/7/2004

German.Cancio@cern.ch
Outline

- ELFms and its subsystems:
  - Quattor
  - Lemon
  - LEAF

- Deployment status
ELFms stands for ‘**Extremely Large Fabric management system**’

**Subsystem:**
- **quattor**: configuration, installation and management of nodes
- : system / service monitoring
- : hardware / state management

- ELFms manages and controls most of the nodes in the CERN CC
  - ~2100 nodes out of ~2400
  - Multiple functionality and cluster size (batch nodes, disk servers, tape servers, DB, web, ...)
  - Heterogeneous hardware (CPU, memory, HD size,..)
  - Supported OS: Linux (RH7, RHEL2.1, RHEL3) and Solaris (9)
http://quatttor.org
Quattor takes care of the *configuration, installation* and *management* of fabric nodes

→ A **Configuration Database** holds the ‘desired state’ of all fabric elements
  - Node setup (CPU, HD, memory, software RPMs/PKGs, network, system services, location, audit info...)
  - Cluster (name and type, batch system, load balancing info...)
  - Defined in templates arranged in hierarchies – common properties set only once

→ Autonomous management agents running on the node for
  - **Base installation**
  - **Service (re-)configuration**
  - **Software installation and management**

→ Quattor was developed in the scope of EU DataGrid. Development and maintenance now coordinated by CERN/IT
Configuration Database

GUI

CLI

Scripts

CDB

RDBMS

XML

Cache

CCM

Node

LEAF, LEMON, others

Node Management Agents

pan

SOAP

HTTP

SQL

Node Management Agents
[lxplus009] ~ > cdbop

CDB CLI: Version 1.7

Enter user-name: gcancio
Enter password:

Connecting to https://cobserv.cern.ch...

Welcome to CDB Command Line Interface
Type 'help' for more info

cdb> open
cdb> get profile_lxb1002
[INFO] getting template: profile_lxb1002.tpl
cdb> list profile_lcgmon*
[INFO] listing templates
profile_lcgmon001d
profile_lcgmon002d

cdb> [BEGIN]
```
[lxplus040] ~ > ncm-query --dump /hardware/harddisks

[INFO] Subtree: /hardware/harddisks
+-harddisks
   +-hda
       $ capacity : (long) '19456'
       $ interface : (string) 'ide'
       $ model : (string) 'WDC WD200BB-00CLB0'

[lxplus040] ~ > ncm-query --dump /hardware/serialnumber

[INFO] Subtree: /hardware/serialnumber
$ serialnumber : (string) '2826000223'

[lxplus040] ~ > ncm-query --dump /system/network/interfaces/eth0

[INFO] Subtree: /system/network/interfaces/eth0
+-eth0
   $ driver : (string) 'e100'
   $ gateway : (string) '137.138.1.1'
   $ ip : (string) '137.138.4.212'
   $ netmask : (string) '255.255.0.0'
```
```sql
SELECT hostname, serialnumber, contracttype
FROM vwhost
WHERE serialnumber IN
(SELECT serialnumber
FROM vwhost
GROUP BY serialnumber
HAVING count(serialnumber) > 1);
```

<table>
<thead>
<tr>
<th>#</th>
<th>HOSTNAME</th>
<th>SERIALNUMBER</th>
<th>CONTRACTTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lxshare0338</td>
<td>2826000109</td>
<td>no contract</td>
</tr>
<tr>
<td>2</td>
<td>lxshare0337</td>
<td>2826000109</td>
<td>no contract</td>
</tr>
<tr>
<td>3</td>
<td>tbcd0040</td>
<td>3040-0061</td>
<td>no contract</td>
</tr>
<tr>
<td>4</td>
<td>tbcd0085</td>
<td>3040-0061</td>
<td>no contract</td>
</tr>
<tr>
<td>5</td>
<td>tbcd0041</td>
<td>3040-0062</td>
<td>no contract</td>
</tr>
<tr>
<td>6</td>
<td>tbcd0086</td>
<td>3040-0062</td>
<td>no contract</td>
</tr>
<tr>
<td>7</td>
<td>tbcd0087</td>
<td>3040-0063</td>
<td>no contract</td>
</tr>
<tr>
<td>8</td>
<td>tbcd0042</td>
<td>3040-0063</td>
<td>no contract</td>
</tr>
</tbody>
</table>
Managing (cluster) nodes

Managed nodes

- SW package Manager (SPMA)
- Cache
- Installed software:
  - kernel, system, applications...
- System services:
  - AFS, LSF, SSH, accounting...

Software Servers

- Software Replicator (SWRep)
- System installer:
  - RH73, RHES, Fedora, ...

Install server

- Vendor
- System installer
- Install Manager
- SW package Manager (SPMA)
- CDB

Installed software:
- kernel, system, applications...

Node Configuration Manager (NCM)

Managed nodes

- Node Configuration Manager (NCM)
- CCM
- RPM, PKG

Install server

- Install Manager
- Node (re)install
- Node Configuration Manager (NCM)
- CDB

Install server

- Install Manager
- Node (re)install
- Node Configuration Manager (NCM)
- CDB

Install server

- Install Manager
- Node (re)install
- Node Configuration Manager (NCM)
- CDB
Node Management Agents

- **NCM (Node Configuration Manager):** framework system, where service specific plug-ins called *Components* make the necessary system changes to bring the node to its CDB desired state
  - Regenerate local config files (e.g. `/etc/sshd/sshd_config`), restart/reload services (SysV scripts)
  - Large number of components available (system and Grid services)

- **SPMA (Software Package Mgmt Agent) and SWRep:** Manage all or a *subset* of packages on the nodes
  - Full control on production nodes: *full control* - on development nodes: *non-intrusive*, configurable management of system and security updates.
  - Package *manager*, not only *upgrader* (roll-back and transactions)

- **Portability:** Generic framework; plug-ins for NCM and SPMA available for RHL (RH7, RHES3) and Solaris 9

- **Scalability to O(10K) nodes**
  - Automated replication for redundant / load balanced CDB/SWRep servers
  - Use scalable protocols eg. HTTP and replication/proxy/caching technology ([slides here](#))
http://cern.ch/lemon
Lemon – LHC Era Monitoring

Lemon

Monitoring Repository

Configuration Database

RRD Tool Framework

Correlation Engine

Node

Monitoring Sensor Agent

Monitoring Sensor

Soap/WSDL

Apache/PHP

Users

Monitoring repository backend

ELFms – German Cancio - n° 15
LEMON

- Monitoring sensors and agent
  - Large amount of metrics (~ 10 sensors implementing 150 metrics)
  - Plug-in architecture: new sensors and metrics can easily be added
  - Asynchronous push/pull protocol between sensors and agent
  - Available for Linux and Solaris

- Repository
  - Data insertion via TCP or UDP
  - Data retrieval via SOAP
  - Backend implementations for text file and Oracle SQL
  - Keeps current and historical samples – no aging out of data but archiving on TSM and CASTOR

- Correlation Engines and ‘self-healing’ Fault Recovery
  - allows plug-in correlations accessing collected metrics and external information (eg. quattor CDB, LSF), and also launch configured recovery actions
  - Eg. average number of users on LXPLUS, total number of active LCG batch nodes
  - Eg. cleaning up /tmp if occupancy > x %, restart daemon D if dead, ... 

- Visualization
  - Next slide

- As with Quattor, LEMON is an EDG development now maintained by CERN/IT
http://cern.ch/leaf
LEAF (LHC Era Automated Fabric): Collection of workflows for *automated* node hardware and state management

- **HMS (Hardware Management System):**
  - Track systems through all steps in lifecycle eg. installation, moves, vendor calls, retirement
  - Automatically requests installs, retires etc. to technicians
  - GUI to locate equipment physically
  - HMS implementation is CERN specific, but concepts and design should be generic

- **SMS (State Management System):**
  - Automated handling high-level configuration steps, eg.
    - Reconfigure and reboot all LXPLUS nodes for new kernel
    - Reallocate nodes inside LXBATCH for Data Challenges
    - Drain and reconfig node X for diagnosis / repair operations
  - Extensible framework – plug-ins for site-specific operations possible
  - Issues all necessary (re)configuration commands on top of quattor CDB and NCM
    - Uses a state transition engine

- **HMS and SMS interface to Quattor and LEMON (or rather: sit on top!)** for setting/getting node information respectively
LEAF screenshots
ELFms status – Quattor (I)

- Manages (almost) all Linux boxes in the computer centre
  - ~ 2100 nodes, to grow to ~ 8000 in 2006-8
  - LXPLUS, LXBATCH, LXBUILD, disk and tape servers, Oracle DB servers
    - Solaris clusters, server nodes and desktops to come for Solaris9

- Starting: head nodes using Apache proxy technology for software and configuration distribution

- Misc developments pending, like
  - Fine-grained ACL protection to templates
  - HTTPS instead of HTTP for CDB profile and SW transport
ELFms status – Quattor (II)

- LCG-2 WN configuration components available
  - Configuration components for RM, EDG/LCG setup, Globus
  - Progressive reconfiguration of LXBATCH nodes as LCG-2 WN’s

- Community driven effort to use quattor for general LCG-2 configuration
  - Coordinated by staff from IN2P3 and NIKHEF
  - Aim is to provide a complete porting of EDG-LCFG config components to Quattor for all LCG services
  - CERN and UAM Madrid providing generic installation instructions and site-independent packaging, as well as a Savannah development portal
    - Installation toolkit, user’s guide, tutorials available

- **EGEE** has chosen quattor for managing their integration testbeds

- **Tier1/2** sites as well as LHC experiments evaluating using quattor for managing their own farms
ELFms status – LEMON (I)

- Smooth production running of MSA agent and Oracle-based repository at CERN-CC
  - 150 metrics sampled every 30s -> 1d
  - ~ 1 GB of monitoring data / day on ~ 2100 nodes
  - New sensors and metrics, eg. tape robots, temperature, SMART disk info
- GridICE project uses LEMON for data collection
- Gathering experiment requirements and interfacing to grid-wide monitoring systems (MonaLisa, GridICE)
  - Good interaction with, and gathered feedback from CMS DC04
  - Archived raw monitoring data will be used for CMS computing TDR
- Visualization:
  - Operators - Test interface to new generation alarm systems (LHC control alarm system)
  - Finish status display pages
ELFms status – LEMON (II)

- Work on redundancy solutions for Monitoring Repository (homegrown and/or Oracle Streams)

- Quality of Service indicators, correlations and actuators (in collaboration with BARC India)
  - Ie. “tell LEAF to reassign two more nodes from LXBATCH to LXPLUS since capacity insufficient”
  - Provide batch job mix indicators for improved I/O and CPU load equilibrium
ELFms status - LEAF

- HMS in full production for all nodes in CC
  - HMS heavily used during CC node migration
- SMS in production for LXBATCH

Next steps:
- Deploy SMS across more clusters
- Tighter HMS/SMS integration (automatic put nodes in and out production during eg. rack moves)

- Developing ‘asset management’ GUI replacing PC finder
  - Client of HMS and SMS
  - Drag&drop nodes to automatically initiate HMS moves
  - Multiple select nodes, then initiate action eg. kernel upgrade
  - Interface to LEMON GUI
Summary

- **ELFms** is deployed in production at CERN
  - Stabilized results from 3-year developments within EDG and LCG
  - Established technology
  - Providing real added-on value for day-to-day operations

- Quattor and LEMON are generic software
  - Other projects and sites getting involved

- Site-specific workflows and “glue scripts” can be put on top for smooth integration with existing fabric environments
  - LEAF HMS and SMS

- **CERN will help with Quattor (and LEMON) deployment at other sites**
  - We provide site-independent software and installation instructions
  - **Collaboration** for providing missing pieces, eg. configuration components, GUI’s, beginner’s user guides?

- More information: [http://cern.ch/elfms](http://cern.ch/elfms)
WP4 architecture concepts

- Information model. Configuration is distinct from monitoring
  - Configuration == desired state (what we want)
  - Monitoring == actual state (what we have)

- Modularity
  - Open interfaces and protocols

- Extensibility
  - Allow for 3rd-party and site specific plug-ins and add-ons

- Scalability
  - Thousands of nodes

- Automation
  - Minimize manual interventions

- Node autonomy
  - Operations are handled locally whenever possible

- Site autonomy
  - A site must keep control of its local resources
The Use of Quattor

*a status report*

Some people from LCG participating institutes took the initiative to develop some essential Quattor modules for the installation, configuration and updates of the LCG2 software suite.

1. First workshop:
   - 8 dedicated testing sites and some others participated
   - In March just after the LCG workshop
   - An critical analysis was made of the usage of LCFGng for the EDG software.
   - Decided on a global configuration schema for the various grid components

2. Priorities:
   - Primarily for LCG2
   - For non-CERN worker nodes initially, then CE, BDII, SE

3. Work done:
   - Some modules written
   - Proper test bed defined and operational

4. Outlook:
   - Expected LCG-2 complete install end of the summer
   - Use in the EGEE JRA1 testing test bed
   - Expect from CERN to keep supporting the Quattor core team
Improvements wrt EDG-LCFG

- New and powerful configuration language
  - True hierarchical structures
  - Extendable data manipulation language
  - (user defined) typing and validation
- SQL query backend
- Portability
  - Plug-in architecture -> Linux and Solaris
- Enhanced components
  - Sharing of configuration data between components now possible
  - New component support libraries
  - Native configuration access API (NVA-API)
- Stick to the standards where possible
  - Installation subsystem uses system installer
  - Components don’t replace SysV init.d subsystem
- Modularity
  - Clearly defined interfaces and protocols
  - Mostly independent modules
  - “light” functionality built in (eg. package management)
- Improved scalability
  - Enabled for proxy technology
  - NFS mounts not necessary any longer
- Enhanced management of software packages
  - ACL’s for SWRep
  - Multiple versions installable
  - No need for RPM ‘header’ files
- Last but not least...: Support!
  - EDG-LCFG is frozen and obsoleted (no ports to newer Linux versions)
  - LCFG -> EDG-LCFGng -> quattor
Differences with ASIS/SUE

ASIS:
- Scalability
  - HTTP vs. shared file system
- Supports native packaging system (RPM, PKG)
- Manages all software on the node
- ‘real’ Central Configuration database
- (But: no end-user GUI, no package generation tool)

SUE:
- Focus on configuration, not installation
- Powerful configuration language
  - True hierarchical structures
  - Extendable data manipulation language
  - (user defined) typing and validation
  - Sharing of configuration data between components now possible
- Central Configuration Database
- Supports unconfiguring services
- Improved dependency model
  - Pre/post dependencies
- Revamped component support libraries
Differences with ROCKS

- Rocks: better documentation, nice GUI, easy to setup
- Design principle: reinstall nodes in case of configuration changes
  - No configuration or software updates on running systems
  - Suited for production? Efficiency on batch nodes, upgrades / reconfigs on 24/24, 7/7 servers (eg. gzip security fix, reconfig of CE address on WN’s)
- Assumptions on network structure (private, public parts) and node naming
- No indication on how to extend the predefined node types or extend the configured services
- Limited configuration capacities (key/value)
- No multiple package versions (neither on repository, nor simultaneously on a single node)
  - Eg. different kernel versions on specific node types
- Works only for RH Linux (Anaconda installer extensions)
NCM Component example

[...]

sub Configure {

    my ($self,$config) = @_;
    # access configuration information
    my $arch=$config->getValue('/system/architecture'); # CDB API
    $self->Fail ("not supported") unless ($arch eq 'i386');
    # (re)generate and/or update local config file(s)
    open (myconfig, '/etc/myconfig'); ...
    # notify affected (SysV) services if required
    if ($changed) {
        system('/sbin/service myservice reload'); ...
    }
}

sub Unconfigure { ... }
Key concepts behind quattor

- Autonomous nodes:
  - Local configuration files
  - No remote management scripts
  - No reliance on global file systems AFS/NFS

- Central control:
  - Primary configuration is kept centrally (and replicated on the nodes)
  - A single source for all configuration information

- Reproducibility:
  - Idempotent operations
  - Atomicity of operations

- Scalability:
  - Load balanced servers, scalable protocols

- Use of standards:
  - HTTP, XML, RPM/PKG, SysV init scripts, ...

- Portability:
  - Linux, Solaris