Quattor : Gérer un Site Complexe

Michel Jouvin

LAL/Orsay

jouvin@lal.in2p3.fr

JOSY, 14/10/08, Montpellier

Agenda

- Qu'est qu'un site complexe ? Les défis ?
- Pourquoi Quattor ?
- Architecture générale et principaux composants
- La description des configurations
- Exemples d'utilisation
  - Mise à jour d'OS
  - Machines virtuelles Xen
- Administration
- L'exemple du GRIF
- Qui utilise Quattor ?
- Conclusions
- Références utiles

Site Complexe ?

- Tout site est complexe
  - Différentes infrastructures à gérer
  - Manpower limité
- Tous les serveurs *presque* similaires…
  - « Presque » est le nœud du problème, pas un début de solution..
  - Solutions de type « clonage » génèrent beaucoup d'exceptions difficiles à gérer
- La complexité peut être liée au grand nombre de machines et/ou leur spécialisation
  - Peu de machines toute différentes peut être aussi difficile à gérer que beaucoup de machines identiques
- Délégation de gestion à l'intérieur d'un site ou entre sites
  - Eviter de dupliquer l'effort entre équipes ou pour 2 configurations gérées par une équipe
- Desktops : concilier gestion centrale et autonomie

Les Défis...

- Factoriser l'effort : éviter de faire deux fois la même chose
  - Faire en sorte que le « coût » d'une opération soit le même pour une ou un grand nombre de machine
  - Optimiser les ressources humaines disponibles
  - Ne pas parcelliser les gens dans des tâches inintéressantes en permettant des « rotations »
- Mutualiser entre sites l'effort de configuration
  - De plus en plus d'infrastructures communes à plusieurs sites (ex: grille)
  - Partager plus que des recettes
- Traçabilité
  - Retrouver l'historique des changements pour comprendre un problème
  - Pouvoir revenir en arrière simplement sur une modification
  - Pouvoir réinstaller à l'identique, y compris config spécifique

… Les Défis

- Vérifier la cohérence des configurations et prédire les effets des changements
  - Ne pas attendre les effets catastrophiques d'un changement pour résoudre un problème
  - Pouvoir décrire la configuration attendue et mettre des contraintes décrivant la cohérence
  - Pouvoir décrire des dépendances entre actions à effectuer et suspendre les opérations dépendant d'une action qui a échoué
- Permettre la reconfiguration à la demande sans réinstallation
  - Même description des configurations lors de l'installation
- Dans un « grand » site, pouvoir contrôler le droit de modifier certaines parties de la configuration
- Permettre la gestion depuis l'environnement préféré des administrateurs
  - OS, GUI ou ligne de commande…

Pourquoi Quattor ?

- Quattor développé par EDG WP4
  - EDG (2001) : premiers besoins de gérer de grandes fermes
  - Installation initiale ET changement de configuration
    - Nécessité d'updates très fréquentes du MW et de sa configuration
  - Outils existants ne traitent que l'un OU l'autre
    - Exemples : Kickstart, Imaging, APT…
  - Expérience du CERN pour gestion de grand nombre de machines : plusieurs générations d'outils internes…
    - Besoin d'une description unique et homogène de toute la configuration
    - Modification atomique des logiciels installés
- Besoin d'avoir des installations reproductibles
  - Réinstaller 1 machine à l'identique suite à une panne
  - Garantir 2 machines identiques pour le même service
- Linux (RH, rpm)
  - Support Solaris a existé mais n'est plus maintenu
- Totalement indépendant du middleware Grille

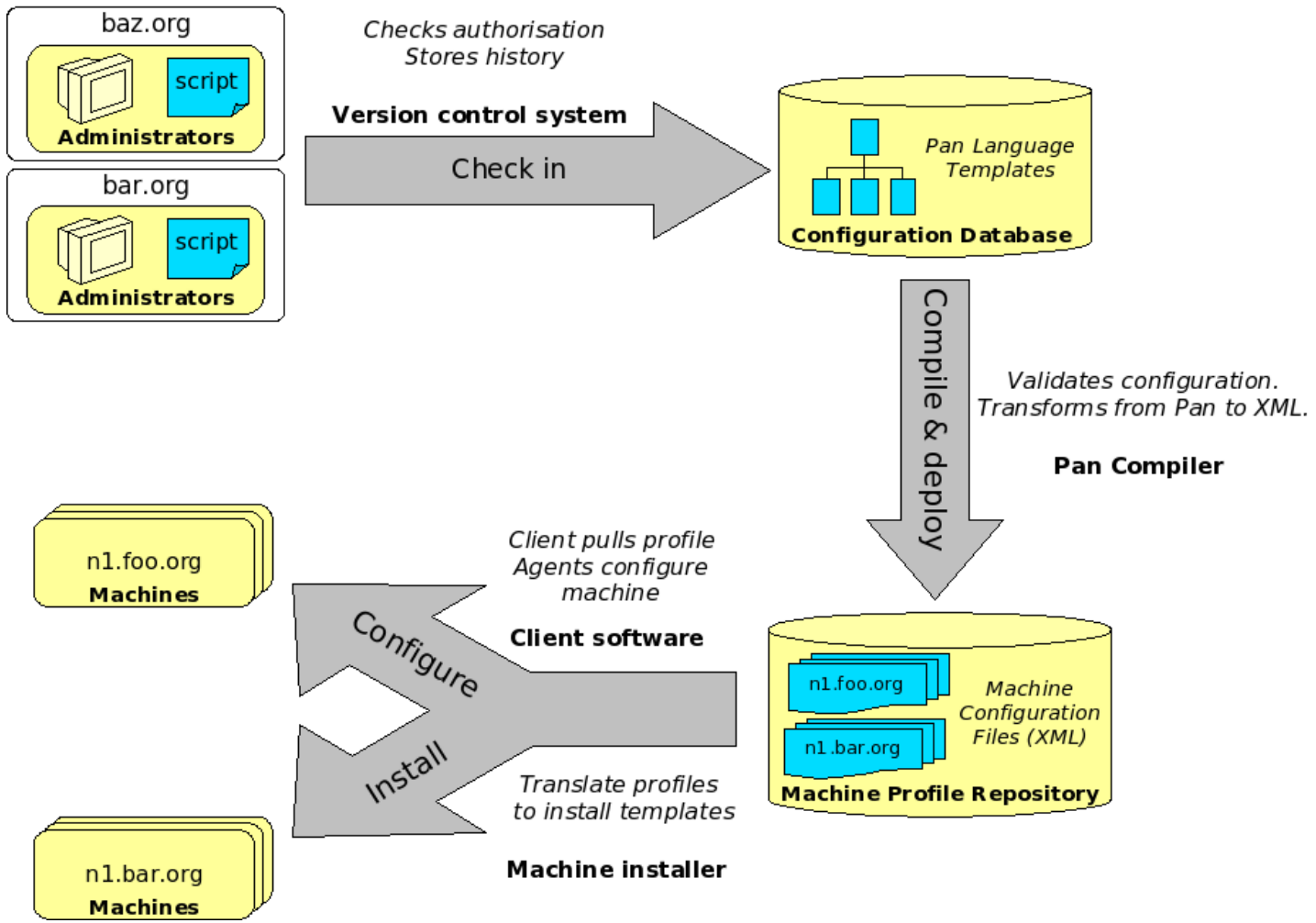
Principales Caractéristiques…

- Toute la configuration d'une machine dans une base de donnée
  - Possibilité d'installation reproductible
  - Modification de la configuration active à partir de la base de données (sans réinstallation)
  - De l'installation initiale à la poubelle…
    - Versionning de la configuration dans la base de donnée
- Factorisation de la production et de la maintenance des configurations
  - Pouvoir produire plusieurs machines *presque* identiques
  - Pouvoir modifier la configuration d'un même service sur des machines de type différent
  - Pouvoir partager des informations de configuration entre machines différentes
    - Ex: configuration réseau liée à un site

… Principales Caractéristiques

- Configuration : description de l'état à atteindre et non du comment
  - Language PAN
- Vérification de la consistance des configurations avant déploiement (compilation)
  - Schéma de l'arbre d'information décrivant la configuration
  - Existence des packages à déployer
- Installation initiale avec l'outil standard de la plateforme configuré à partir de la base de donnée
  - Kickstart pour RH / SL
- Gérer des configurations « multi-sites »
  - Depuis un « point » unique ou non
  - Avec ou sans hierarchie de délégation (« devolved management »)
  - Contrôle des droits à changer certaines parties de la configuration

Quattor Workflow



Quattor (Core Components)…

- Base de donnée de configuration basée sur Subversion
  - SCDB basé sur des logiciels standards : SVN + ant
  - « Target » ant : code (Java) spécifique à Quattor
- Configuration décrite avec des templates
  - Description de la configuration finale, pas du comment
    - Langage PAN
  - Héritage des templates
  - Hardware et software
- Serveur de déploiement : dhcpd + tftpd
  - Serveur est notifié d'un changement dans la base de donnée de configuration
  - Notifie les clients affectés par les changements (pas de login)
  - Possibilité de plusieurs serveurs attachés à la même base de donnée (configuration multi-site)
  - Aucun composant spécifique à Quattor

… Quattor (Core Components)

- Client : Node Configuration Manager (NCM)
  - Récupère les informations de configuration
  - Fait les actions nécessaires pour atteindre la configuration demandée
  - Basé sur le concept de composants (plugins)
    - 1 composant gère un aspect de la configuration (NIS, accounts…)
    - 1 composant est un script Perl avec une API standard
    - Plus d'une centaine disponible : 50% services standard, 50% pour services grilles gLite
    - Facilement extensible en écrivant de nouveaux composants
- Management possible depuis Unix, Windows et Mac
  - Compilateur PAN (Java) + client SVN + ant disponibles sur toutes les plateformes

Quattor (Optionel)…

- Déploiement d'application : SPMA
  - Implémenté comme un composant NCM
  - Développé dans le cadre de Quattor
    - Utilise une liste explicite des packages et versions à installer
    - Permet la modification atomique de logiciels (rpmt)
    - Permet upgrade et downgrade
    - Vérifie les dépendances entre logiciels
  - Aussi possible d'utiliser APT ou YUM mais très peu utilisé
    - N'offre pas le contrôle des versions installées et la possibilité de rollback
- Software Repository : HTTPrep
  - Contient les RPMs à installer via SPMA
  - Accessible/accédé via HTTP
  - Target ant permettant la génération des templates PAN utilisés par SPMA
    - Possibilité d'utiliser YUM pour prédire et aider à résoudre les problèmes de dépendance lors de la génération des templates
  - Possibilité d'utiliser Squid pour répliquer le serveur de RPM

… Quattor (Optionel)

- Installation initiale : AII
  - Permet l'installation initiale du système et du client Quattor
    - Configuration et déploiement d'applications via NCM
  - Utilise l'installer standard (Kickstart/Anaconda) et PXE
    - Configuration DHCP, TFTP, KS à partir de la base de données de configuration
  - Possibilité de lancer une réinstallation sans accès à la console
    - Machines toujours configurées pour booter en PXE
    - Sélection boot local, réinstallation ou rescue par l'outil de gestion AII (aii-shellfe)

Configuration et Templates

- Configuration d'une machine est décrite dans un fichier XML : profil
  - 1 profil par machine
  - Décrit le HW présent, le SW à installer (RPMs) et la configuration du SW
    - Possibilité de configurer le SW à partir du HW présent
  - Seule information utilisée par les autres composants Quattor (NCM, AII…)
- Profil construit à partir d'une description de haut niveau : *templates*
  - Templates utilisent un langage particulier : PAN
  - Templates sont compilés pour produire le profil (panc)
  - 1 profil peut être constitué à partir d'un grand nombre de templates
    - Templates similaires à des « poupées russes »
  - Des templates standards pour OS et MW gLite EGEE, ainsi que des exemples pour de nombreux services

Language PAN…

- Langage déclaratif
  - Décrit des états à atteindre
  - Permet la manipulation de données : calcul d'une valeur à partir d'une autre, listes, validation de valeur…
  - Non procédural : impossible de spécifier des actions ou de définir des blocs conditionnels
    - Assignation conditionnelle
- Possibilité de définir des fonctions
  - Ecrites en langage PAN
  - Manipulation et calcul des données
  - Fonction de validation : exécutée à la fin de la compilation pour vérifier la conformité des valeurs
- Assignement : classique ou valeur par défaut
  - Permet à un site de redéfinir une valeur par défaut avant la configuration d'un service

… Language PAN

- Organisation hiérarchique de l'information
  - Ressource : peut contenir des ressources ou des propriétés
  - Propriété : attribut ayant une valeur
  - Exemple :
    - « /hardware/network/interfaces/eth0/ip » = « 134.158.88.34 »;
- Typage fort des données
  - Définition des types (schéma) de données de l'ensemble de l'arbre d'information
    - Similaire à une définition de structure/classe
    - 1 définition standard proposée, actuellement utilisée dans des contextes très variés
  - Fonctions de conversion entre type
- Possibilité de définir des variables
  - Utilisables comme paramètres des autres définitions
  - Permettent de définir des templates génériques

Exemple de Templates...

- Tous les profils pour 1 type de machine sont identiques (même si les machines sont différentes)

```
object template profile_ipnls2005;

include machine-types/wn;

# Add repositories
include repository/config;
```

- Type WN basé sur d'autres templates standards configurant les différents services

```
template machine-types/wn;
…
# VO configuration
variable CONFIGURE_VOS = true;
variable CREATE_HOME ?= undef;
variable NODE_VO_PROFILE_ENV = true;

# Include base configuration of a gLite node
include { 'machine-types/base' };

# Include WN components
include { 'glite/wn/service' };

# Add site specific configuration, if any
include { return(WN_CONFIG_SITE) };

…
```

… Exemple de Templates

- Tous utilisent la définition de base d'une machine gLite

```
unique template machine-types/base;
…
# profile_base for profile structure
include { 'quattor/profile_base' };

# NCM core components
include { 'components/spma/config' };
include { 'components/grub/config' };

# hardware
include { 'hardware/functions' };
"/hardware" = if ( exists(DB_MACHINE[escape(FULL_HOSTNAME)]) ) {
                  create(DB_MACHINE[escape(FULL_HOSTNAME)]);
              } else {
                  error(FULL_HOSTNAME + " : hardware not found in machine database");
              };

# common site machine configuration
include { 'pro_site_config' };

# File system configuration.
variable FILESYSTEM_LAYOUT_CONFIG_SITE ?= "site/filesystems/glite";
variable FILESYSTEM_CONFIG_SITE ?= "site/filesystems/glite";

# Select OS version based on machine name
include { 'os/version' };

# Load gLite version information
include { 'defaults/glite/version' };

…
```

Exemple : Paramètres de Sites

- Spécificités d'un site à l'aide de variable sans modifications des templates génériques (95%)

```
template site/glite/config;
…
# MYPROXY CONFIGURATION ------------------------------------------------

variable PX_HOST   ?= "myproxy.grif.fr";
variable GRID_TRUSTED_BROKERS ?= list(
  "/O=GRID-FR/C=FR/O=CNRS/OU=LAL/CN=grid09.lal.in2p3.fr",
  "/O=GRID-FR/C=FR/O=CEA/OU=DAPNIA/CN=node27.datagrid.cea.fr",
  "/O=GRID-FR/C=FR/O=CEA/OU=IRFU/CN=node08.datagrid.cea.fr",
);

# MON BOX PARAMETERS (R-GMA), Apel
variable MYSQL_PASSWORD ?= "wp6_saclay";
variable MON_HOST  ?= "node06.datagrid.cea.fr";
variable APEL_ENABLED ?= true;
variable APELDB_PWD ?= "lhc4G";

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

variable LFC_HOSTS  ?= nlist(
  "grid14.lal.in2p3.fr",       nlist('alias', 'lfc.grif.fr'),
);
variable GRIDICE_SERVER_HOST ?= MON_HOST;

# RB / WMS -------------------------------------------------------------

variable RB_HOST   ?= "node04.datagrid.cea.fr";
variable WMS_HOST   ?= "wms.grif.fr";
…
```

Les Principales Opérations

- Ensemble des opérations via des cibles ‘ant’
  - Principe similaire à ‘Make’
  - Entièrement écrit en Java
    - gestion possible depuis toutes les plateformes
- 4 principales opérations
  - Edition des templates
    - Pour les opérations courantes, surtout les paramètres du site ou définition de nouveaux types de machine (combinaison de services)
  - ant compile : compile (localement) **tous** les profils affectés par les modifications
    - Gestion des dépendances, compilation incrémentale
  - ant deploy : déploie les nouveaux profils
    - Compilation des profils sur le serveur, notification des clients
    - Vérifie que toutes les modifications sont dans le repository (commit) et que l’espace de travail est à jour par rapport au repository
  - ant update.rep.templates : regénère le template associé à chaque repository

Mettre à jour OS

- Récupérer les templates standard de la version
  - RPMs requis : générés par des scripts qui analysent la description fournie par RH (comps.xml)
- Définition de la nouvelle version à utiliser pour la ou les machines
  - En général un template par site : site/os/version_db.tpl
  - ‘grid10.lal.in2p3.fr’, ‘sl460-x86_64’
- `ant compile` + svn commit + ‘ant deploy’
  - Recompile les profils
  - Notifie les clients qu’un nouveau profil est disponible
- Similaire pour le MW de grille gLite
  - Configuration du MW : templates génériques maintenus par le LCG QWG

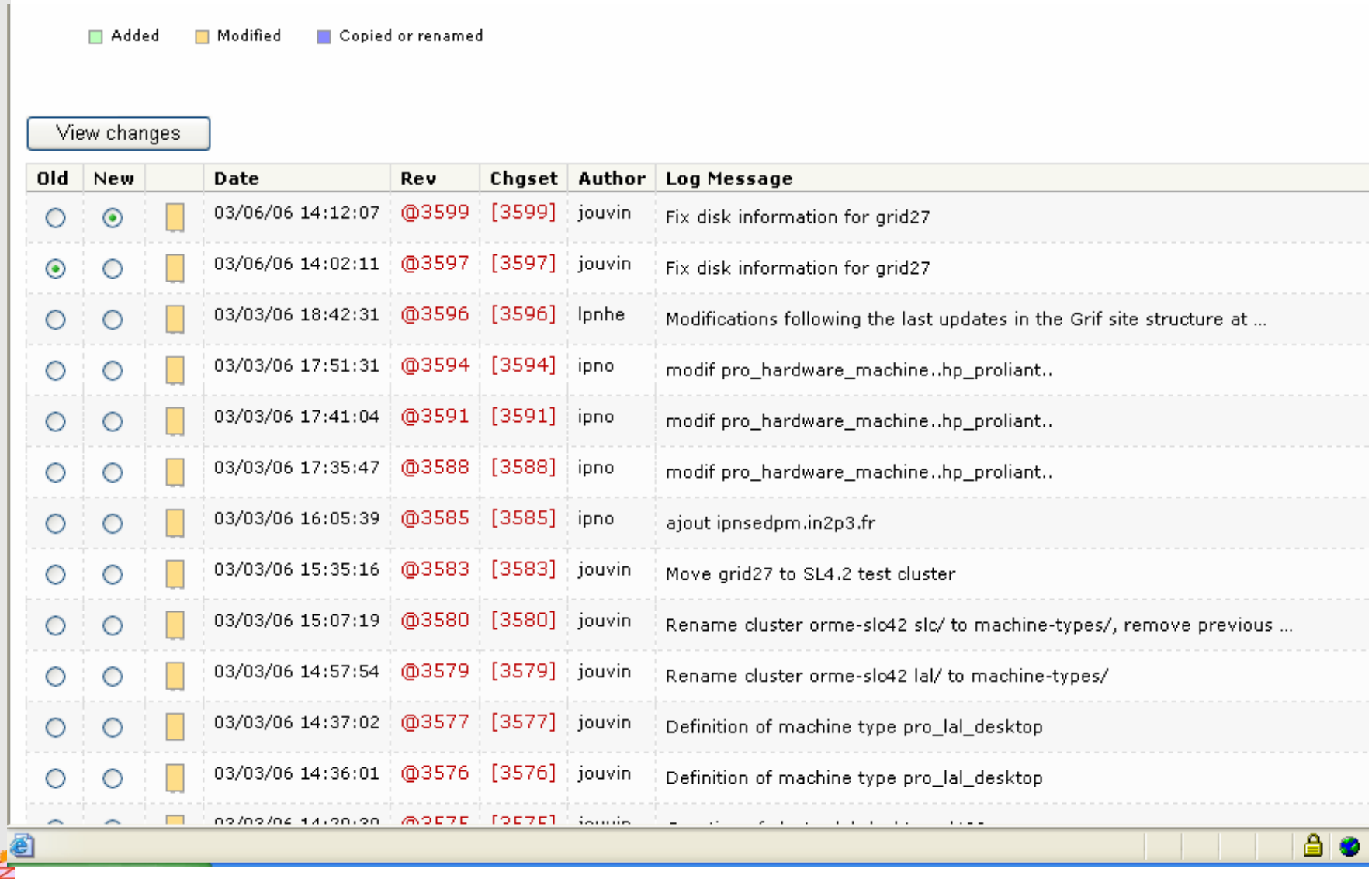
Gestion des Machines Virtuelles

- Dans les templates standard pour les VMs Xen
- Chaque VM est gérée comme une machine réelle
  - 1 HW template générique, sauf MAC address
  - Applique l'ensemble de la méthodologie d'installation et de gestion, y compris l'installation initiale
    - Boot PXE utilise pypxeboot
- Dom0 : quelques lignes pour définir la liste des VMs
  - Toute la configuration faite par Quattor, y compris création des volumes LVM pour disque de boot…
  - Exemple :

```
# Configure Xen and one VM (WN)
variable XEN_DOM0_MEM ?= '3G';
#variable XEN_GUESTS = list("gridwn2.cci.ucad.sn");
variable XEN_VG ?= 'vg.01';
include { 'xen/host/config' };
```

- Documentation : https://trac.lal.in2p3.fr/LCGQWG/wiki/Doc/OS/Xen

SCDB : Outil de Suivi...



Added Modified Copied or renamed

View changes

Old	New		Date	Rev	Chgset	Author	Log Message
○	◉		03/06/06 14:12:07	@3599	[3599]	jouvin	Fix disk information for grid27
◉	○		03/06/06 14:02:11	@3597	[3597]	jouvin	Fix disk information for grid27
○	○		03/03/06 18:42:31	@3596	[3596]	lpnhe	Modifications following the last updates in the Grif site structure at ...
○	○		03/03/06 17:51:31	@3594	[3594]	ipno	modif pro_hardware_machine..hp_proliant..
○	○		03/03/06 17:41:04	@3591	[3591]	ipno	modif pro_hardware_machine..hp_proliant..
○	○		03/03/06 17:35:47	@3588	[3588]	ipno	modif pro_hardware_machine..hp_proliant..
○	○		03/03/06 16:05:39	@3585	[3585]	ipno	ajout ipnsedpm.in2p3.fr
○	○		03/03/06 15:35:16	@3583	[3583]	jouvin	Move grid27 to SL4.2 test cluster
○	○		03/03/06 15:07:19	@3580	[3580]	jouvin	Rename cluster orme-slc42 slc/ to machine-types/, remove previous ...
○	○		03/03/06 14:57:54	@3579	[3579]	jouvin	Rename cluster orme-slc42 lal/ to machine-types/
○	○		03/03/06 14:37:02	@3577	[3577]	jouvin	Definition of machine type pro_lal_desktop
○	○		03/03/06 14:36:01	@3576	[3576]	jouvin	Definition of machine type pro_lal_desktop

… SCDB : Outil de Suivi

Changeset 3568

Timestamp: 03/03/06 11:14:27
Author: jouvin

Message: Use an OS version independent template to add openafs client; define auger1 as an xtremweb server

Files:
- trunk/cfg/clusters/lal-sl420/profiles/profile_auger1.tpl (1 diff)
- trunk/cfg/os/sl305-i386/os/pro_os_openafs_client.tpl
- trunk/cfg/os/sl420-i386/os/pro_os_openafs_client.tpl
- trunk/cfg/sites/lal/machine-types/pro_lal_config_afs_client.tpl (1 diff)

Unmodified | Added | Removed | Modified | Copied | Moved

trunk/cfg/clusters/lal-sl420/profiles/profile_auger1.tpl

r3516	r3568	
10	10	
11	11	`define variable XW_STARTUP_START = false;`
12		`#include pro_lal_server_physics_xtremweb;`
13		`include pro_lal_server;`
	12	`include pro_lal_server_physics_xtremweb;`
14	13	
15	14	

trunk/cfg/sites/lal/machine-types/pro_lal_config_afs_client.tpl

r3371	r3568	
12	12	`define variable PKG_ARCH_KERNEL_MODULE_OPENAFS = PKG_ARCH_KERNEL;`
13	13	

Terminé

L'exemple de GRIF

- GRIF : « gros » site EGEE en région IdF
  - Réparti sur 6 sites (laboratoires IN2P3 + CEA/IRFU)
  - 600+ machines, 5 MSI2K, 500 TB
  - http://grif.fr
- 1 seul site grille géré par une équipe de 20 personnes réparties dans les différents laboratoires
  - Beaucoup d'administrateurs à temps partiel (10 FTE)
  - Degrés divers d'expertise grille
  - 1 réelle équipe technique : 1 réunion mensuelle F2F + communication quotidienne (email principalement)
  - 1 outil collaboratif : Trac. Wiki + Issue Tracker + client SVN
  - Déploiement d'une modif affectant toutes les nodes : ~5mn
- Toute la configuration commune
  - Paramètres dépendant du site (ex: réseau) définis par site
  - DB contient aussi des machines non grille spécifique à chaque site (serveurs locaux, machines virtuelles)

Qui utilise Quattor ?

- Utilisation en augmentation
  - 50+ sites, principalement Europe, généralement sites EGEE
    - Plusieurs groupes de site gérés depuis 1 seule database
  - Plusieurs pays “quattorisés” pour leurs infrastructure grille
    - France, Espagne, Irelande, Belgique
- Grande variété de taille
  - CERN : ~5000 nodes
  - Plusieurs LCG T1s : NIKHEF, CNAF, PIC
  - Beaucoup de LCG T2s et T3s, certains gros (GRIF, DESY, 500+ nodes) et d’autres plus modestes (<50 nodes)
- Principale utilisation est la gestion des ressources grille…
  - Quattor a été conçu pour ça…
- Mais un intérêt grandissant pour les autres ressources
  - Serveurs internes, desktops, virtual machines (XEN)
  - Morgan & Stanley : choix de Quattor (vs. Puppet) à cause du langage PAN, remplacement de SCDB par Git+home tools

Conclusions

- La gestion de tout site est complexe
  - Croissance du nombre de machines partout
  - Diversité des services
  - Manpower faible par rapport aux missions à assumer
- La qualité des outils est déterminante pour faire face à la complexité des tâches ASR
  - Doivent permettre à la fois une gestion cohérente, centralisée (pour éviter les différences inutiles) et souple pour prendre en compte les variantes
  - Doivent permettre la délégation de l'administration
- Quattor est un outil unique utilisé principalement dans le contexte des nœuds de grille mais adapté à de nombreux autres besoins
  - Souplesse et simplicité pour la gestion des VMs
- La possibilité de mutualisation inter-site est un atout majeur
  - Partager des recettes et des exemples n'est pas suffisant

Documentation et Références

- Le site QWG : http://trac.lal.in2p3.fr/LCGQWG
  - Description des templates standards
  - Description de l'utilisation pour la grille
  - Description SCDB
  - Description du langage PAN
  - Exemples fonctionnels
- Le site officiel : http://quattor.org
  - En cours de migration sur SourceForge
- Devolved Management of Distributed Infrastructures with Quattor – LISA08
  - http://www.usenix.org/events/lisa08/tech/
- PLUME : en cours de publication…