

Projet PLACO

Benoit Métrot
benoit.metrot@math.univ-poitiers.fr

UMR 7348 - Laboratoire de Mathématiques et Applications (Poitiers)

Rencontres Mathrice
Caen, mars 2013



Plan

- 1 Historique
- 2 Principe de fonctionnement
- 3 Développement actuel et futur
- 4 Utilisation de PLACO au LMA
- 5 Conclusion

Progression

- 1 Historique
- 2 Principe de fonctionnement
- 3 Développement actuel et futur
- 4 Utilisation de PLACO au LMA
- 5 Conclusion

Idée initiale

Mise à disposition pour une communauté scientifique d'une plateforme collaborative prête à l'emploi mais néanmoins personnalisable

- *Juillet 2008* : Réponse à l'appel à projet de la MRCT → soutien financier de la MRCT
- *2 Décembre 2008* : Rencontre avec différents responsables de réseau de la MRCT (Renatis, Réseau des mécaniciens), étude du besoin
- *Debut 2009* : Constitution d'une équipe de développeurs
- *Avril 2009* : Début du développement

Projet MRCT

- *septembre 2009* : Premier prototype
- *Octobre 2009* : Présentation à la journée JoSy "Outils collaboratifs"
- *Decembre 2009* : Poster à JRES
- *Mars 2010* : Rencontres Mathrice - TP mise en oeuvre PLACO
- *Juin 2010* : ANGD Placo (utilisateurs / administrateurs)

Et après...

- Poursuite du développement
- Exploration de nouvelles pistes (distributions supplémentaires)
- Support auprès des collègues effectuant des essais
- Mise en oeuvre d'un Placo pour le Laboratoire Euro-Maghrébin de Mathématiques et leurs Interactions

Progression

- 1 Historique
- 2 Principe de fonctionnement**
- 3 Développement actuel et futur
- 4 Utilisation de PLACO au LMA
- 5 Conclusion

Principe

- Transforme un serveur en une plateforme collaborative
- Organisation modulaire autours de machines virtuelles
- Spécification des besoins et paramètres dans des fichiers textes simples
- Installation automatisée des outils collaboratifs
- Plateforme prête à l'emploi en quelques minutes

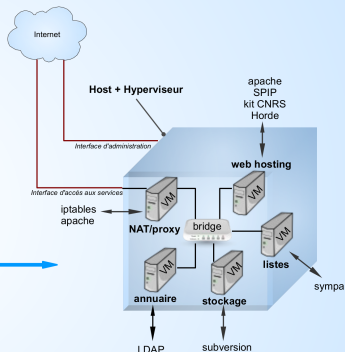
Plateforme PLACO en mars 2011

Générer une plate-forme en 6 commandes

```
$ wget -P /tmp https://svn.math.cnrs.fr/placodev/tags/stable/install.sh
$ ./tmp/install.sh
$ placosh init_platform
$ placosh install web_hosting
$ placosh install lists
$ placosh install svn
```

avec 3 commandes → 1 host + 1 hyperviseur + 1 virtual network
+ NAT / reverse proxy (VM)
+ annuaire (VM)

avec 3 commandes de plus → hébergement web (VM)
+ listes de diffusion (VM)
+ stockage / gestion de version (VM)



Fonctionnement

- **Fichiers `.defines`**

Définissent les propriétés des VMs et de la plateforme

- Ensemble de scripts appelés en cascade

- ▶ `toplevel` : action de haut niveau (créer une VM)
- ▶ `middleware` : action intermédiaire (configurer le réseau)
- ▶ `template` : action élémentaire (installer un paquet)

- Générateur de scripts *placomgr*

Produit des scripts (ou fichiers de configuration) à partir des templates et de fichiers `.defines` pour les déployer ou les exécuter sur les VM



Fonctionnement

- Fichiers `.defines`

Définissent les propriétés des VMs et de la plateforme

- Ensemble de scripts appelés en cascade

- ▶ `toplevel` : action de haut niveau (créer une VM)
- ▶ `middleware` : action intermédiaire (configurer le réseau)
- ▶ `template` : action élémentaire (installer un paquet)

- Générateur de scripts *placomgr*

Produit des scripts (ou fichiers de configuration) à partir des templates et de fichiers `.defines` pour les déployer ou les exécuter sur les VM



Fonctionnement

- Fichiers `.defines`

Définissent les propriétés des VMs et de la plateforme

- Ensemble de scripts appelés en cascade

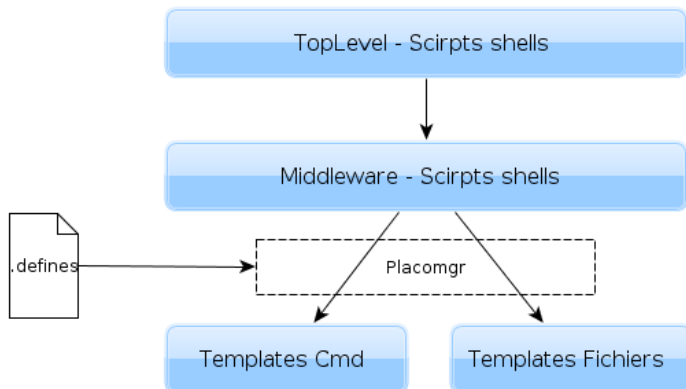
- ▶ `toplevel` : action de haut niveau (créer une VM)
- ▶ `middleware` : action intermédiaire (configurer le réseau)
- ▶ `template` : action élémentaire (installer un paquet)

- Générateur de scripts *placomgr*

Produit des scripts (ou fichiers de configuration) à partir des templates et de fichiers `.defines` pour les déployer ou les exécuter sur les VM



Modèle en couches



Fichiers .defines

```
#
# sample.defines : definition of the default platform
# $Id: default.defines-dist 1096 2011-06-24 13:52:31Z metrot $
#
<network ecole>
  address 10.234.80.1
  netmask 255.255.255.0
  network 10.234.80.0
  broadcast 10.234.80.255
  <vm_address>
    dns 10.234.80.2
  </vm_address>
</network>

dns1_address 194.167.50.187
dns2_address 10.234.80.2
dns_domain ecole.prive

include install.defines
include localhost.defines
```



Deux types de templates

- Template fichier

- ▶ Pour créer un fichier à partir d'un modèle
- ▶ Interprétation du code entre délimiteurs ({} ! et !)
- ▶ `placomgr put resolv.conf /etc/resolv.conf -m vm`
- ▶ Copie vers la VM cible par SCP

- Template de commandes

- ▶ Pour un script de commandes à exécuter
- ▶ Interprétation du code entre délimiteurs ({} ! et !)
- ▶ `placomgr rexec restart_postfix.ex -m vm`
- ▶ `placomgr lexec config_vm.ex -m localhost`
- ▶ Exécution distante du script de commandes par SSH sur machine cible

→ Spécification par distribution/système de virtualisation

Exemple avec un template fichier : resolv.conf

```
{! # $Id$ !}  
{! # maintainer: Benoit Metrot !}  
{! # comment: DNS resolver configuration !}  
# {!$TEMPLATE!}: processed by {!$ENV{USER}!} with {!$0!} from  
    {!$_ARGV[0]!}  
domain {!$dns_domain!}  
search {!$dns_domain!}  
{! if (defined($dns1_address)) { $OUT.=sprintf("nameserver %s\n",  
    "$dns1_address"); } !}  
{! if (defined($dns2_address)) { $OUT.=sprintf("nameserver %s\n",  
    "$dns2_address"); } !}
```



Exemple avec un template fichier : resolv.conf

Fichier copié sur la VM par SCP

```
# template/resolv.conf: processed by root with /placo/lib/  
    placomgr from  
domain ecole.prive  
search ecole.prive  
nameserver 194.167.50.187  
  
nameserver 10.234.80.2
```



Progression

- 1 Historique
- 2 Principe de fonctionnement
- 3 Développement actuel et futur**
- 4 Utilisation de PLACO au LMA
- 5 Conclusion

Orientations

Séparation des couches :

- Virtualisation :
 - ▶ Gestion des VMs (création, destruction)
 - ▶ Installation automatisée des distributions (système minimal)
 - ▶ Paramétrages communs (DNS, Messagerie, Routage)
- Services et Applications :
 - ▶ Outils : Horde Groupware, SVN, Sympa, Sites Web
 - ▶ Installation, paramétrage et intégration

Ce qui est implémenté

- Gestion de machines virtuelles
 - ▶ Création de VM Xen ou KVM
 - ▶ Installation automatisée Debian ou CentOS
 - ▶ Installation des VMs par le réseau ou DVD
 - ▶ Interconnexion des VMs (adressage réseau, multi vlan)
 - ▶ Configuration des services de base (DNS, mail root, mises à jour)
- Support des distributions
 - ▶ GNU/Linux Debian 6.0 avec Xen
 - ▶ GNU/Linux Debian 6.0 avec Kvm
 - ▶ CentOS 6 avec Kvm

Ce qu'il reste à faire

- Intégrer l'installation des services collaboratifs
- Documenter, en détails, le fonctionnement (graphes d'appels)
- Analyseur syntaxique pour la vérification des `.defines`
- Migrer les deux sites web actuels (SPIP + TRAC) vers la forge de la PLM

Progression

- 1 Historique
- 2 Principe de fonctionnement
- 3 Développement actuel et futur
- 4 Utilisation de PLACO au LMA**
- 5 Conclusion

PLACO comme plateforme de virtualisation

Apports pour la gestion VM :

- Automatisation des installations de base
- Standardisation des installations
- Intégration aux services réseau
 - ▶ Messagerie
 - ▶ Supervision
 - ▶ Sauvegarde
 - ▶ Gestion des fichiers journaux
 - ▶ ...

PLACO comme plateforme de virtualisation

Apports pour l'infrastructure :

- Abstraction de la technologie de virtualisation
- Facilite le déplacement des VMs
- Rapidité de reconstruction des VMs
- Personnalisation des configurations

Déploiement

- Installation d'un serveur avec une Debian minimale
- Création d'une configuration initiale (Fichiers `.defines`)
- Script `placoupdate.sh`
 - ▶ Dans `/placo`, installation des scripts, templates distribués par le projet
 - ▶ Ajouts de templates spécifiques au laboratoire
 - ▶ Dépôt de la configuration dans `/placo/etc`
- Initialisation de la plateforme

```
bash$> cd /placo
bash$> /placo/bin/placosh
bash$> init_platform
```

- Création de VM : `create_vm.sh -m mavm`

Progression

- 1 Historique
- 2 Principe de fonctionnement
- 3 Développement actuel et futur
- 4 Utilisation de PLACO au LMA
- 5 Conclusion**

Conclusion

- 5 plateformes en production (environ 20 VM) :
 - ▶ 3 x Debian GNU/Linux 6.0 avec KVM
 - ▶ 1 x Debian GNU/Linux 6.0 avec Xen
 - ▶ 1 x CentOS 6.2 avec Kvm
- Impulser une nouvelle dynamique dans l'équipe *placo-dev*
- Poursuite du développement (OwnCloud ?)