

Parcs hétérogènes, configurations logicielles hétérogènes et mouvantes : une solution à faible coût humain

Frédéric Bloise

École Supérieure d'Ingénieurs de Luminy (ESIL), Université de la Méditerranée
f@esil.univ-mrs.fr

Gérard Milhaud

École Supérieure d'Ingénieurs de Luminy (ESIL), Université de la Méditerranée
Centre Informatique et Systèmes de Communication d'Aix-Marseille (CISCAM)
g@esil.univ-mrs.fr

Résumé

Nous présentons très rapidement les étapes du développement de notre solution de déploiement. Elle est fondée sur l'outil Rembo Toolkit connecté à une base MySQL et s'administre depuis un navigateur web quelconque. Le lecteur intéressé trouvera beaucoup plus de détails à : http://www.esil.univ-mrs.fr/lafirme/je_deploie_dans_la_joye.

Mots clefs

Gestion de parc, déploiement, Rembo Toolkit, Windows, Linux, MySQL

1 Motivation

On le sait, les administrateurs systèmes sont souvent en situation de sous-effectif. Ils ont bien du mal à absorber les tâches quotidiennes sans grand intérêt pour pouvoir s'atteler à des projets intéressants de plus long terme [1]. Tout l'enjeu est donc de gagner du temps sur ces tâches récurrentes. Nous nous intéressons ici à l'une des plus prenantes : la gestion d'un parc informatique important (150 à 200 machines par membre du service) hétérogène aussi bien au niveau matériel que logiciel.

2 La problématique de la gestion de parc

Elle se décompose en plusieurs sous tâches :

1. (Ré)installation du système et des logiciels
2. Réparation logicielle du système ou d'un logiciel au comportement dégradé
3. Mise à jour système et logiciels, ajout/suppression de logiciels
4. Maintenance des matériels

Nous proposons ici une solution visant à gagner du temps du point de vue logiciel, donc sur les points 1, 2 et 3.

En effet, ces opérations sont récurrentes et accaparantes à l'extrême en particulier dans les cas de type UFR, où les configurations logicielles sont très mouvantes : incessantes demandes d'installation de nouveaux logiciels ; salles de formation à profil logiciel changeant d'une semaine sur l'autre, etc.

A ceci s'ajoute souvent une grande hétérogénéité matérielle du parc que l'avènement des systèmes d'exploitation Windows 2000 et ultérieurs rend encore plus problématique : l'OS est si intimement attaché au matériel qu'il est impossible de réutiliser l'image clonée d'un disque dur sur une autre architecture.

Et pour finir, on s'aperçoit que cet énorme travail est la plupart du temps reproduit à l'identique autant de fois qu'il y a de composantes autonomes dans sa structure (Université, groupes de laboratoires, etc.).

Ces contraintes nous ont amenés à définir le cahier des charges minimal qu'à notre sens, toute «bonne» solution de déploiement doit respecter. Ce fut notre base de développement.

3 Le cahier des charges

- L'approche devra être valide pour au moins nos deux systèmes cibles principaux : Linux et Windows 2000 et ultérieurs,
- l'installation du système d'exploitation ne doit se faire qu'une fois par architecture différente, et si possible à partir d'une image disque unique quasi-finalisée, au matériel près,
- l'installation d'un logiciel ne doit se faire qu'une seule fois par système d'exploitation, et permettre de produire un paquetage réutilisable,

- les paquetages doivent pouvoir s’exporter facilement, indépendamment du type de matériel, afin d’être mutualisés entre les diverses composantes de la structure,
- l’installation d’une machine doit pouvoir être programmée à la carte, en puisant librement dans le réservoir de paquetages,
- le comportement de la machine au démarrage doit pouvoir être conditionné par les valeurs de variables récupérées aussi bien par analyse matérielle de la machine qu’au travers de ressources réseaux quelconques,
- il doit être possible d’utiliser la notion de groupes de machines afin de leur associer des profils thématiques, et de particulariser une machine au sein d’un groupe en surchargeant le profil du groupe pour cette machine,
- il doit être possible de programmer des interfaces afin que réinstallation et choix de la configuration logicielle puisse être faits sans compétences informatiques,
- tout le développement autour du produit choisi devra pouvoir s’opérer à l’aide d’outils standards libres et gratuits,
- le coût financier par machine de l’outil choisi doit être raisonnable.

4 Notre solution

4.1 L’outil logiciel

Très vite, au vu des contraintes posées, l’outil `Rembo Toolkit` de la société Rembo s’est imposé face aux solutions classiques et limitées de clonage de disques. Tirant pleinement partie de la technologie PXE (il nécessite d’ailleurs des cartes réseaux compatibles PXE), il permet de «scripter» le comportement de la machine avant le démarrage de l’OS, de créer des paquetages logiciels, et intègre la notion de groupe de machines.

4.2 L’image de base auto-adaptative

Notre premier travail a consisté à créer une image générique de toutes nos architectures sous `Linux`, incluant tous les modules associés aux divers composants matériels de nos machines. Au démarrage, le script `Rembo` va modifier le fichier adéquat afin d’activer uniquement les modules correspondant à la machine.

Pour `Windows 2000/XP`, nous avons utilisé une procédure consistant à éliminer d’une image de l’OS finalisée toute particularité matérielle. Ainsi, pour une nouvelle architecture, nous installons cette image et `Windows` détecte le nouveau matériel et l’installe. Reste l’installation du matériel non reconnu par `Windows` et nous disposons d’une image pour la nouvelle architecture à peu de frais.

4.3 La création des paquetages

Pour chaque OS, nous créons ensuite les paquetages logiciels au fur et à mesure des besoins : il deviennent ensuite disponibles pour chaque machine, indépendamment de son architecture matérielle.

4.4 Notion de groupes de machines

Par salle et/ou par architecture, nous créons des groupes de machines disposant d’un même profil logiciel, c’est-à-dire une image de base de l’OS et une liste de logiciels. Chaque machine du groupe hérite de ce profil et dispose de plus d’un profil personnel qui vient surcharger le profil du groupe.

4.5 Connexion à une base `MySQL` et interface `WEB`

Après une première approche où l’ensemble du système était géré à partir des variables `Rembo`, associées aux groupes et aux machines, nous avons déplacé la quasi-totalité de l’information contenue dans les variables `Rembo` à l’intérieur d’une base de données `MySQL` d’inventaire des machines. Cette base est gérée via le web par des pages `PHP`, et les scripts `Rembo` attaquent la base. Les avantages sont nombreux ; nous disposons enfin d’un produit qui correspond à nos aspirations :

- Plus besoin de connaissance du système `Rembo` pour gérer le parc, une couche d’abstraction a été ajoutée. Une page web par machine, par groupe et par salle permet de modifier le profil logiciel associé : on coche ou décoche les logiciels et au prochain démarrage, la machine consultera la base et se conformera à son nouveau profil.
- Délégation aisée : page dédiée avec mot de passe pour tel groupe de machines, telle salle, plus besoin de donner l’accès à la console `Rembo`.
- Consultation aisée via le web par les utilisateurs (enseignants, étudiants) des logiciels disponibles avec l’assurance de disposer d’une information à jour.

Références

- [1] Gérard Milhaud et Olivier Pagé. Petit manuel anti-dépression à l’usage des administrateurs systèmes et réseaux. Dans *Actes de la conférence JRES2001*, pages 227–242, Lyon, Décembre 2001.