



Gestion de parc simplifiée grâce aux systèmes de paquets

Julien BLACHE

<jblache@debian.org>

Projet Debian

Solutions Linux 2004 Paris – 4 Février 2004

Sommaire



- État des lieux
- Boîte à outils
 - Déploiement
 - Maintenance
- Paquets Debian
 - Créer un paquet
 - Modifier un paquet
 - Effectuer un backport
- Documentation

États des lieux

- Plusieurs outils d'installation ou de déploiement
 - Plus élégants que le clonage
 - Plus flexibles
 - Parfois plus rapides
 - Possibilité de paralléliser les installations plus facilement et plus efficacement
- Peu, voire pas, d'outils de maintenance satisfaisants
 - Recours à des solutions individuelles

Conséquences : administrateur livré à lui-même, écriture de scripts plus ou moins fiables, risques d'accidents, ...

Constat : Debian plutôt bien lotie, en partie grâce à un bon système de paquets.

Outils : installation

- FAI – *Fully Automatic Installation*
 - Spécifique à Debian
 - Se base sur le système de paquets
 - Transferts de données réduits
 - Espace de stockage réduit sur le serveur
 - Rapidité
 - Flexibilité (système de classes)
- System Imager
 - Générique à tous les systèmes GNU/Linux
 - Uniquement sur i386 et IA64 pour le moment

Futur : fonctionnalité d'installation automatisée intégrée à `debian-installer`

Outils : maintenance générique

- System Imager
 - Mise à jour d'une machine avec un golden master
 - Coûteux : long, transferts de données importants
 - Solution totale : système + configuration
- cfengine
 - Gestion de la configuration d'un parc de machines
 - Complexe, mais efficace une fois maîtrisé
- dsh - *Distributed SHell*
 - Exécution en parallèle ou en série à distance (ssh)

Outils : maintenance Debian

- **dpkg + aptitude**
 - Synchronisation des paquets installés
 - `dpkg --get-selections > liste.txt`
 - `cat liste.txt | dpkg --set-selections`
 - `aptitude install`
- **debconf + LDAP**
 - Stockage de la base debconf sur un serveur LDAP
- **kernel-package**
 - Empaquetage du kernel
- **apt-proxy, debmirror, apt-ftpparchive**
 - Économie de bande passante
 - Création d'un miroir local partiel
 - Création d'une archive locale

Créer un paquet : composition

Un paquet Debian = fichiers de contrôle + fichiers du logiciel

Fichiers de contrôle :

- Informations de dépendance, de version, etc...
- Scripts d'installation et de désinstallation
- Liste des fichiers de configuration
- Checksums du contenu du paquet

Fichiers du logiciels :

- Binaires, données, etc...

Valeur ajoutée du paquet : les fichiers de contrôle, une configuration de base qui marche avec des valeurs par défaut saines.

Créer un paquet : principe

Procédure simplifiée :

- Récupérer les sources du logiciel
- Décompresser, nettoyer les sources
- Lancer `dh_make`
- Adapter les scripts et fichiers de contrôle
- Construire le paquet

Tout l'art de la manoeuvre se situant à l'étape d'adaptation des scripts et fichiers de contrôle...

Créer un paquet : le répertoire debian/

- `control` : dépendances, description
- `changelog` : version, changelog
- `copyright` : informations de copyright et licence
- `rules` : script de construction
- `postinst`, `postrm`, ... : scripts de maintenance
- `docs` : liste de la documentation du logiciel
- `README.Debian` : informations spécifiques au paquet
- `templates`, `config` : interface `debconf`
- `patches/` : patches appliqués à la construction
(`dpatch`, `dbs`, ...)

Créer un paquet : paquet source

Dans le cas d'un paquet normal :

- `paquet_1.2.orig.tar.gz`
 - Archive source upstream
 - Si possible non modifiée
- `paquet_1.2-3.diff.gz`
 - Modifications Debian
 - Répertoire `debian/`
- `paquet_1.2-3.dsc`
 - Fichier de contrôle
 - MD5SUMS des fichiers précédents
 - Signé avec la clé GPG du mainteneur

Dans le cas d'un paquet spécifique Debian (*Debian native*) :

- `paquet_1.2.3.tar.gz`
- `paquet_1.2.3.dsc`

Créer un paquet : outils

- `pbuilder` : constructeur de paquets en environnement minimal
- `debhelper` : collection de scripts d'aide à la construction de paquets
- `dh-make` : utilitaire d'aide à la *debianisation*, fournit un squelette de paquet source
- `devscripts` : collection de scripts utilitaires pour la construction de paquets
- `lintian` et `linda` : vérificateurs de paquets

Créer un paquet : conseils

- Se conformer à la Debian Policy
 - Gage d'intégration et de pérennité
- Documenter les modifications
- Tester le paquet
 - Installation / désinstallation
 - Réinstallation (avec ou sans purge préalable)
 - Retour à une version précédente
- Possibilité d'utiliser des méta-paquets
 - Paquet vide dépendant d'un ensemble de paquets
 - Installation d'un ensemble de paquets par le jeu des dépendances

Démonstration : création d'un paquet



Création d'un paquet

Modifier un paquet : principe

Procédure :

- Récupérer les sources
 - Automatiquement avec apt
`apt-get source paquet`
 - Manuellement, puis décompression par
`dpkg-source -x fichier.dsc`
- Installer les dépendances de compilation
`apt-get build-dep paquet`
- Effectuer les modifications nécessaires
- Construire le paquet

Modifier un paquet : conseils

Deux règles d'or pour modifier un paquet :

- Documenter les modifications effectuées
- Suffixer la révision Debian du paquet
 - Identificateur local
 - Révision locale
 - Exemple : 1.2-3 devient **1.2-3.0jb1**

But :

- Identifier les paquets modifiés en cas de rapport de bug chez Debian
- Empêcher `apt` de réinstaller la version Debian
- Permettre le passage à une nouvelle version Debian

Backport : principe

Backport : ou «rétro-portage», reconstruire pour stable un paquet pris dans unstable.

Procédure :

- Récupérer les sources
 - Automatiquement avec apt
`apt-get source paquet`
 - Manuellement, puis décompression par
`dpkg-source -x fichier.dsc`
- Vérifier les dépendances de compilation
- Installer, au besoin backporter, ces dépendances
- Effectuer les modifications nécessaires
- Construire le paquet

Backport : pièges

- Attention aux logiciels écrits en C++
 - Changement d'ABI C++ entre Woody et Sarge/Sid
- Attention à l'environnement de compilation
 - Strict minimum requis pour la construction du paquet
 - Utilisation d'un chroot `pbuilder` fortement conseillée

Backport : conseils

- Comme pour les paquets modifiés localement, il est conseillé de suffixer la version du paquet avec un identifiant local accompagné d'une révision locale.
- Encore plus que pour un paquet modifié, test intensif du paquet obtenu
- Ne pas dupliquer les efforts :
<http://www.fs.tum.de/~bunk/packages/>
<http://apt-get.org>

Documentation – Ressources

- FAI
<http://www.informatik.uni-koeln.de/fai/>
- System Imager
<http://www.systemimager.org>
- debconf (paquet debconf-doc)
`/usr/share/doc/debconf-doc`
- Paquets Debian
#debian-devel-fr sur irc.debian.org
<http://www.debian.org/devel>
 - New Maintainers' Guide
 - Developers' Reference
 - Debian Policy Manual

Questions/Réponses



Slides téléchargeables (PDF)

<http://people.debian.org/~jblache/events/sl2004/talks>