Pourquoi et comment faire sa première contribution à Debian ?

Lucas Nussbaum
lucas@debian.org

Moi

- Ingénieur ENSIMAG 2005
- Doctorant en fin de thèse, enseignant à l'IUT2 de Grenoble (bientôt à l'univ. Lyon 1)
- Utilisateur de Debian depuis 2002
- Actif dans son développement depuis 2005
- Entré dans NM en septembre 2005, DD depuis fin 2006

Dans Debian:

- Co-maintenance de l'interpréteur Ruby et de bibliothèques Ruby
- Quality Assurance

Introduction

Debian

- Plus grosse distribution communautaire
- Partie intégrante de l'histoire du Logiciel Libre

Chiffres

- Bientôt 15 ans
- 12000 paquets source
- 20000 paquets binaires
- Une dizaine d'architectures supportées

Introduction

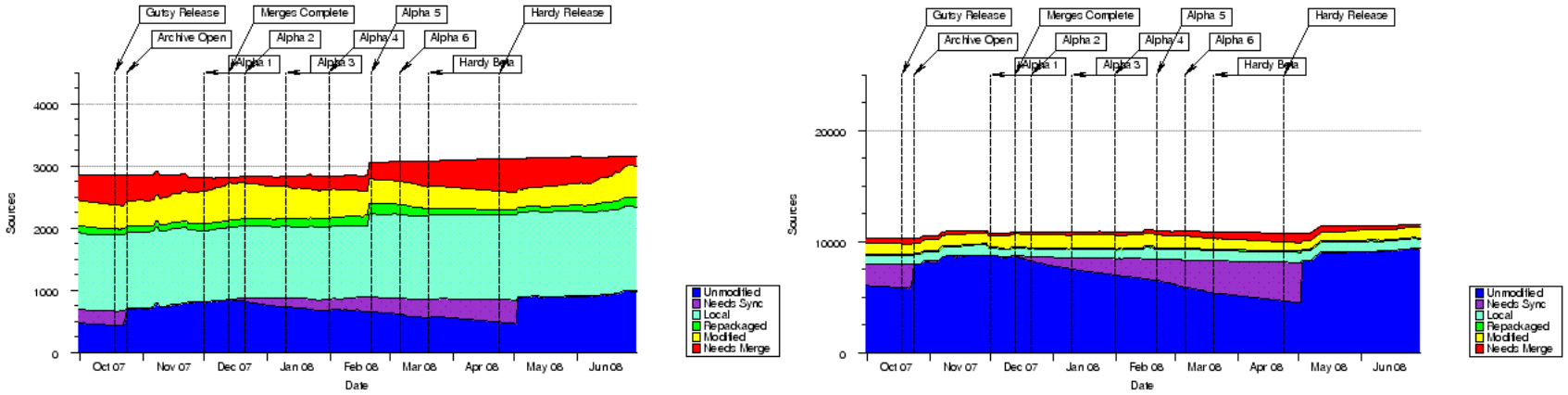
- Une communauté vivante et indépendante
- environ 600 développeurs actifs, avec des expertises variées
- Base de nombreux succès

... mais pourquoi vous ?

- Vous pouvez faire une différence
- Vous avez (forcément) des besoins spécifiques
- Debian a besoin de vous
  - De nombreux paquets sont pas/mal maintenus
  - De nombreuses équipes sont surchargées
- Participer à Debian est devenu plus facile
  - Développement organisé en équipes (environ 80 équipes différentes) - `http://wiki.debian.org/Teams`
  - Debian Maintainers

... et pourquoi pas Ubuntu ?

- Communauté d'utilisateurs très active (mais un peu jeune)
- Processus moins carré que Debian
- Moins de notion de propriété de paquets (compétences plus dilluées)
- Pour la plupart des paquets, le vrai travail est fait dans Debian. Ubuntu se contente de gérer la divergence.



- La plupart des décisions sont prises par Canonical
- La vraie question: acceptez-vous de travailler (gratuitement) pour Canonical ?

Comment faire son premier patch ?

Objectif : Donner les bases vous permettant de modifier un paquet

Ceci n'est pas un cours de Packaging !

Documents utiles:

- Debian New Maintainer's Guide
- Debian Developers Reference
- Debian Policy

Paquets utiles: `build-essential, devscripts`

Attention: quelques simplifications (volontaires !)

Paquet source, paquet binaire

1 paquet source génère 1 ou plusieurs paquets binaires

1 paquet binaire peut fonctionner sur toutes les architectures (`Architecture: all`) ou devoir être compilé sur chaque architecture par les *buildds*.

Paquet source

- fichier .dsc, et d'autres fichiers
- cas le plus courant:
  - `cowsay_3.03-9.1.dsc`: fichier de description
  - `cowsay_3.03.orig.tar.gz`: archive upstream
  - `cowsay_3.03-9.1.diff.gz`: différences par rapport à upstream
- manipulation avec `dpkg-source`
- récupérer un paquet source:
  `apt-get source cowsay` ou
  `dget http://somewhere/cowsay_3.03-9.1.dsc`.

Répertoire `debian/`

- `control`
- `changelog` (installé dans `/usr/share/doc/<paquet>/changelog.Debian.gz`)
- `rules`
- et d'autres !

debian/control

```
Source: cowsay
Section: games
Priority: optional
Maintainer: Franz Pletz <fpletz@franz-pletz.org>
Build-Depends: debhelper (>= 5)
Homepage: http://www.nog.net/~tony/warez/cowsay.shtml
Standards-Version: 3.7.3

Package: cowsay
Architecture: all
Depends: ${perl:Depends}
Recommends: filters
Description: A configurable talking cow
 Cowsay (or cowthink) will turn text into happy ASCII cows, with
 speech (or thought) balloons. If you don't like cows, ASCII art is
 available to replace it with some other creatures (Tux, the BSD
 daemon, dragons, and a plethora of animals, from a turkey to
 an elephant in a snake).
```

debian/rules

- Makefile
- (Presque) personne ne l'écrit à la main
- Helpers: `debhelper`, `cdbs`

debhelper et cdbs

Debhelper:

- Commandes `dh_*` permettant de simplifier les tâches courantes.
- Exemples:
  - `dh_installchangelogs ChangeLog`
  - `dh_compress`
  - `dh_md5sums`
- controlables via arguments, ou via fichiers spécifiques dans debian/

CDBS:

- Ensemble de classes pour Make automatisant l'appel à `dh_*` (dans le bon ordre !)
- Rend `debian/rules` très (trop ?) simple

Apporter des modifications

- **nécessaire** (bugfixes, intégration)
- 2 solutions:
  - Modifier directement le source upstream
  - Utiliser un *patch system*

Patch systems

- `simple-patchsys.mk` (cdbs), `dpatch`, **`quilt`**
- Permet de séparer et de documenter chaque changement fait au source upstream
- Patches dans `debian/patches/` (généralement)
- `debian/rules patch/unpatch`

Construire ("builder") un paquet

- `dpkg-buildpackage`
- Génère les paquets binaires pour votre architecture
- Pour installer les *build-dependencies*:
  `apt-get build-dep cowsay`
- Pour construire dans un environnement propre: `pbuilder`

Exporter les changements

- `debdiff`
- sur les fichiers .dsc: voir les changements entre les paquets source
- sur les fichiers .changes: voir aussi les changements entre les paquets binaires

Et ensuite ?

- Envoyer le patch sur le BTS (mail à `nnnnnn@bugs.debian.org`)
- Rejoindre une équipe de maintenance
- RTFM :-)
- Corriger des bugs critiques
- Adopter un paquet orphelin, ou packager un nouveau logiciel
- Trouver un sponsor
- Devenir Debian Maintainer
- Devenir DD
- Aller dans l'espace
- Changer le monde

`lucas@debian.org`