

DÉMYSTIFIER



Café Guix mardi 30 janvier 2024

Yann Dupont

Nicolas Vallet

Yann Dupont <Yann.Dupont@univ-nantes.fr>, Nicolas Vallet <nls.vallet@gmail.com>

TORDONS LE COU AUX MYTHES...

- (Malheureusement parfois entendu) :
 - « GUIX c'est peut-être bien
 - sans doute pour la science reproductible...
 - ... mais je n'en ai pas l'usage»
 - « GUIX est trop compliqué
 - Surtout pour "mes" utilisateurs.
 - Ils ont besoin de trucs simples. »
 - Docker/singularity/conda/appimage/snap/flatpak/modules ça fait pareil
 - « Il n'y a pas beaucoup de logiciels»
 - « Ça n'est pas aussi bien que ce que je compile à la main »

DÉBUTANTS BIENVENUS ! (SESSION 1 GNU)

- Pour ce qui suit :
 - Distribution utilisée : commune, mais ancienne : Ubuntu 18.04
 - Guix est déjà installé (Par magie ? détail à suivre !)
 - Le compte utilisateur « moi » est vide.

```
moi@Scarab:~$ python3
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[GCC 8.4.0] on linux
```

```
moi@Scarab:~$ gcc -v
Using built-in specs.
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
```

- Distribution ancienne, plus à jour.

DÉBUTANTS BIENVENUS ! (CEINTURE VERTE)

```
moi@Scarab:~$ guix
guix : nom de commande manquant
Essayez « guix --help » pour plus d'informations.
```

```
moi@Scarab:~$ guix --help
Utilisation : guix COMMANDE ARGS...
Lance la COMMANDE avec les arguments ARGS.
```

COMMANDE doit être une des sous-commandes listées ci-dessous :

```
commandes principales
[...]
install  install packages 1
package  manage packages and profiles
pull     pull the latest revision of Guix
remove   remove installed packages 1
[...]
```

1 Rappelle apt, yum

(Ce café se focalisera en partie sur ces parties « basiques »).

GUIX, LE NOM DES PAQUETAGES

Comment trouver le nom des paquetages à installer ?

```
moi@Scarab:~$ guix search gcc
name: gcc-toolchain
version: 9.3.0
[...]
name: gcc-toolchain
version: 8.4.0
[...]
name: gcc-toolchain
version: 10.2.0
outputs: out debug static
systems: x86_64-linux i686-linux
dependencies: binutils@2.34 gcc@10.2.0 glibc@2.31 ld-wrapper@0
[...]
```

Alternativement,

```
moi@Scarab:~$ /var/guix/profiles/per-user/guixuser/current-guix/bin/guix package --list-available=gcc
[...]
gcc-toolchain 9.3.0 out,debug,static gnu/packages/commencement.scm:3836:4
gcc-toolchain 8.4.0 out,debug,static gnu/packages/commencement.scm:3836:4
[...]
gcc-toolchain 10.2.0 out,debug,static gnu/packages/commencement.scm:3836:4
```

gcc-toolchain représente le nom du paquetage logiciel. Sans autre précision, la dernière version (10.2.0 ici) sera considérée.

Alternativement, aller sur <https://packages.guix.gnu.org/>

HTTPS://PACKAGES.GUIX.GNU.ORG/



This package browser is in early development. Mind the rough edges.

Home → Packages

Packages

GNU Guix provides 27,165 packages transparently available as pre-built binaries. These pages provide a complete list of the packages. Our [continuous integration system](#) shows their current build status (updated January 29, 2024).



libreoffice 7.6.3.1

Office suite

texlive-onrannual 66594

Class for Office of Naval Research
Ocean Battlespace Sensing annual
report

ghc-texmath 0.12.5.4

Conversion between formats used to
represent mathematics

librevenge 0.0.4

Document importer for office suites

oauth2ms 0.0.0-0.a1ef0ca

XOAUTH2 compatible Microsoft
Office 365 token fetcher

GUIX, GESTIONNAIRE DE PAQUETS

```
moi@Scarab:~$ guix install python gcc-toolchain
guix install: warning: Votre installation Guix est vieille de 1059 jours.
guix install: warning: Envisagez d'exécuter « guix pull » suivi de
« guix package -u » pour obtenir des paquets à jour et des mises à jour de sécurité.
```

Les paquets suivants seront installés :

```
gcc-toolchain 10.2.0
python        3.8.2
```

- (Les installations prennent un certain temps)
- Finit par rendre la main, après l'affichage d'un conseil.

 Ce n'est pas plus compliqué qu'apt ou yum !.

conseil : Pensez à paramétrer les variables d'environnement nécessaires en lançant :

```
GUIX_PROFILE="/home/moi/.guix-profile"
."$GUIX_PROFILE/etc/profile"
```

Autrement, regardez `guix package --search-paths -p "/home/moi/.guix-profile"`.

(Comme tout conseil, nous allons royalement l'ignorer...)

TESTS

```
moi@Scarab:~$ python3
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[...]
gcc -v
[...]
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
```

- Ça ne fonctionne pas ?
 - Il est temps de relire le fameux conseil !
- Exécuter :

```
GUIX_PROFILE="/home/moi/.guix-profile"
."$GUIX_PROFILE/etc/profile"
```

- Ensuite

```
moi@Scarab:~$ python3
Python 3.8.2 (default, Jan 1 1970, 00:00:01)
[...]
moi@Scarab:~$ gcc -v
[...]
gcc version 10.2.0 (GCC)
```

COMMENT EST-CE POSSIBLE ?

- Il n'y a pas eu à passer root ?
- Où sont installés ces logiciels ?

```
moi@Scarab:~$ which python3
/home/moi/.guix-profile/bin/python3
moi@Scarab:~$ which gcc
/home/moi/.guix-profile/bin/gcc
moi@Scarab:~$
```

- Dans le home ?

```
moi@Scarab:~$ du -s .
104.
```

Impossible d'avoir gcc et python3 dans si peu d'espace.

```
moi@Scarab:~$ ls -al /home/moi/.guix-profile
lrwxrwxrwx 1 moi moi 44 janv. 26 10:43 /home/moi/.guix-profile -> /var/guix/profiles/per-user/moi/guix-profile
moi@Scarab:~$ ls -al /var/guix/profiles/per-user/moi/guix-profile
lrwxrwxrwx 1 moi moi 19 janv. 26 10:51 /var/guix/profiles/per-user/moi/guix-profile -> guix-profile-1-link
moi@Scarab:~$ ls -al /var/guix/profiles/per-user/moi/guix-profile-1-link
lrwxrwxrwx 1 moi moi 51 janv. 26 10:51 /var/guix/profiles/per-user/moi/guix-profile-1-link -> /gnu/store/08zj827a8pnw142n37ps135cdlhwyxw1-profile
moi@Scarab:~$ ls -al /gnu/store/08zj827a8pnw142n37ps135cdlhwyxw1-profile
```

LE PROFIL

```
ls -al /gnu/store/08zj827a8pnw142n37ps135cdlhwyxw1-profile/bin
total 256
dr-xr-xr-x 1 root root 1172 janv. 1 1970 .
dr-xr-xr-x 1 root root 134 janv. 1 1970 ..
lrwxrwxrwx 1 root root 65 janv. 1 1970 2to3 -> /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2/bin/2to3
lrwxrwxrwx 1 root root 69 janv. 1 1970 2to3-3.8 -> /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2/bin/2to3-3.8
lrwxrwxrwx 1 root root 78 janv. 1 1970 addr2line -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/addr2line
lrwxrwxrwx 1 root root 71 janv. 1 1970 ar -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/ar
lrwxrwxrwx 1 root root 71 janv. 1 1970 as -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/as
lrwxrwxrwx 1 root root 72 janv. 1 1970 c++ -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/c++
lrwxrwxrwx 1 root root 78 janv. 1 1970 catchsegv -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/catchsegv
lrwxrwxrwx 1 root root 76 janv. 1 1970 c++filt -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/c++filt
lrwxrwxrwx 1 root root 72 janv. 1 1970 cpp -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/cpp
lrwxrwxrwx 1 root root 77 janv. 1 1970 easy_install-3.8 -> /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2/bin/easy_install-3.8
lrwxrwxrwx 1 root root 76 janv. 1 1970 elfedit -> /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0/bin/elfedit
```

- gcc et python3 sont respectivement stockés dans

`/gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2`

et

`/gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2`



Chaque paquetages est stocké dans sa propre arborescence.

DÉPENDANCES

```
moi@Scarab:~$ ldd /home/moi/.guix-profile/bin/python3
linux-vdso.so.1 (0x00007ffc30fa5000)
libpython3.8.so.1.0 => /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2/lib/libpython3.8.so.1.0 (0x000079bf89200000)
libcrypt.so.1 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libcrypt.so.1 (0x000079bf89678000)
libpthread.so.0 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libpthread.so.0 (0x000079bf89657000)
libdl.so.2 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libdl.so.2 (0x000079bf89652000)
libutil.so.1 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libutil.so.1 (0x000079bf8964d000)
libm.so.6 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libm.so.6 (0x000079bf890bf000)
libexpat.so.1 => /gnu/store/imh5xxqw10dql4crlngbbjh4r24raf4j-expat-2.2.9/lib/libexpat.so.1 (0x000079bf89617000)
libgcc_s.so.1 => /gnu/store/01b4w3m6mp55y531kyi1g8shh722kwqm-gcc-7.5.0-lib/lib/libgcc_s.so.1 (0x000079bf895fe000)
libc.so.6 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libc.so.6 (0x000079bf88f02000)
/gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/ld-linux-x86-64.so.2 (0x000079bf896b4000)
moi@Scarab:~$
```

- Il n'y a pas d'utilisation des bibliothèques natives système (Ici, Ubuntu)
- Tout est cohérent et stocké dans /gnu/store
- Le logiciel est portable (et fonctionnera aussi sous Debian, Redhat, Suse...)

SOURCER L'ENVIRONNEMENT GUIX

Se déloger, se reloger

```
moi@Scarab:~$ python3
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
moi@Scarab:~$ which python3
/usr/bin/python3
```

À nouveau le python de la distribution (Ubuntu)! Faut-il tout réinstaller ?

```
moi@Scarab:~$ ls -l .guix-profile
lrwxrwxrwx 1 moi moi 44 janv. 26 10:43 .guix-profile -> /var/guix/profiles/per-user/moi/guix-profile
```

Non, le profil Guix est toujours présent.

```
moi@Scarab:~$ source ~/.guix-profile/etc/profile
moi@Scarab:~$ python3
Python 3.8.2 (default, Jan 1 1970, 00:00:01)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

- L'environnement Guix n'est pas automatiquement chargé.
- Le contenu du profil vient se superposer à celui du système.

GUIX PULL

Lors du premier install :

```
guix install: warning: Votre installation Guix est vieille de 1059 jours.  
guix install: warning: Envisagez d'exécuter « guix pull » suivi de  
« guix package -u » pour obtenir des paquets à jour et des mises à jour de sécurité.
```

D'où vient le binaire guix ?

```
moi@Scarab:~$ guix --version  
guix (GNU Guix) 0bd58ae7ff0d5c3935b0d660819aca4a0493ef8a  
Copyright © 2021 les auteurs de Guix  
[...]  
moi@Scarab:~$ which guix  
/usr/local/bin/guix
```

Le binaire guix a été installé sur le système (+ là-dessus ensuite)

```
moi@Scarab:~$ guix pull  
Mise à jour du canal « guix » depuis le dépôt Git « https://git.savannah.gnu.org/git/guix.git »...  
réception des objets 29% [#####] 1
```

1 (Le processus est assez long -plusieurs minutes-, surtout lorsque le binaire guix est ancien, comme ici)

construction du profil avec 1 paquet...
conseil : Pensez à paramétrer les variables d'environnement nécessaires en lançant :

```
GUIX_PROFILE="/home/moi/.config/guix/current"  
."$GUIX_PROFILE/etc/profile"
```

Autrement, regardez `guix package --search-paths -p "/home/moi/.config/guix/current"`.

GUIX PULL

Exécution de ce qui est demandé :

```
moi@Scarab:~$ GUIX_PROFILE="/home/moi/.config/guix/current"  
moi@Scarab:~$ . "$GUIX_PROFILE/etc/profile"
```

```
guix --version  
guix (GNU Guix) 467192552859f8637a83304c1b101ed440de24e3  
Copyright © 2024 les auteurs de Guix  
[...]  
moi@Scarab:~$ which guix  
/home/moi/.config/guix/current/bin/guix
```

- La version de la commande guix est à jour et est désormais **personnelle** !
- Tout est stocké dans le store global (/gnu/store).
- Toujours pas eu besoin de passer root.
- guix pull fait beaucoup plus qu' apt update . La commande prend du temps, elle **génère** une nouvelle version de Guix. Ne le faire que lorsque c'est nécessaire.
- Le binaire guix ainsi généré est auto-suffisant, il contient la liste des paquetages disponibles et leurs dépendances.

GUIX UPGRADE

guix install: warning: Envisagez d'exécuter « guix pull » suivi de « guix package -u » pour obtenir des paquets à jour et des mises à jour de sécurité.

Nb. `guix package -u` et `guix upgrade` sont synonymes.

```
guix upgrade
```

```
Les paquets suivants seront mis à jour :
```

```
gcc-toolchain 10.2.0 → 13.2.0  
python       3.8.2 → 3.10.7
```

```
La dérivation suivante sera compilée :
```

```
/gnu/store/jcl3l2dd7brg71acnnyzrv9ignpgjqsq-profile.drv
```

```
construction de l'ensemble des certificats des AC...
```

```
[...]
```

```
construction du profil avec 2 paquets...
```

```
conseil : Pensez à paramétrer les variables d'environnement nécessaires en lançant :
```

```
GUIX_PROFILE="/home/moi/.guix-profile"
```

```
moi@Scarab:~$ python3
```

```
Python 3.10.7 (main, Jan 1 1970, 00:00:01) [GCC 11.3.0] on linux
```

```
[...]
```

```
moi@Scarab:~$ gcc -v
```

```
[...]
```

```
gcc version 13.2.0 (GCC)
```

- Les logiciels sont bien plus à jour.
- Il faut bien recharger le profil (`. "$GUIX_PROFILE/etc/profile"`)
- Où est-ce stocké ?

LES PROFILS

```
moi@Scarab:~$ which python3
/home/moi/.guix-profile/bin/python3
moi@Scarab:~$ ls -ald /home/moi/.guix-profile
lrwxrwxrwx 1 moi moi 44 janv. 26 10:43 /home/moi/.guix-profile -> /var/guix/profiles/per-user/moi/guix-profile
moi@Scarab:~$ ls -al /var/guix/profiles/per-user/moi/guix-profile
lrwxrwxrwx 1 moi moi 19 janv. 26 17:37 /var/guix/profiles/per-user/moi/guix-profile -> guix-profile-2-link
```

Il y a une question de génération de profils ! De fait une commande existe :

```
moi@Scarab:~$ guix package -l
Génération 1 26 janv. 2024 10:51:35
python      3.8.2 out /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2
gcc-toolchain 10.2.0 out /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0

Génération 2 26 janv. 2024 17:37:17 (actuelle)
+ gcc-toolchain 13.2.0 out /gnu/store/i3x1cr4b0f20r8pkcfq7sxmwwv6261b0y-gcc-toolchain-13.2.0
+ python      3.10.7 out /gnu/store/8s8k2v6b5b7lyf5y5n534cjd9sipgas6-python-3.10.7
- gcc-toolchain 10.2.0 out /gnu/store/pknm43xsza6nlc7bn27djip8fis92akd-gcc-toolchain-10.2.0
- python      3.8.2 out /gnu/store/fmdscjlqdz5pb6ijlqfi7kmwc0a634a-python-3.8.2
```

Chaque version coexiste dans le store global /gnu/store et de façon indépendante, avec sa propre arborescence (et ses propres dépendances)

DÉPENDANCES (SUITE)

```
moi@Scarab:~$ ldd /home/moi/.guix-profile/bin/python3
linux-vdso.so.1 (0x00007fffde9b0000)
libpython3.10.so.1.0 => /gnu/store/8s8k2v6b5b7lyf5y5n534cjd9sipgas6-python-3.10.7/lib/libpython3.10.so.1.0 (0x00007111e1e00000)
libcrypt.so.1 => /gnu/store/ln6hxqjvz6m9gdd9s97pivlqck7hzs99-glibc-2.35/lib/libcrypt.so.1 (0x00007111e22bf000)
libm.so.6 => /gnu/store/ln6hxqjvz6m9gdd9s97pivlqck7hzs99-glibc-2.35/lib/libm.so.6 (0x00007111e21e2000)
libexpat.so.1 => /gnu/store/94rdaacvsqf05zhw88w92y8bkvgxdfpl-expat-2.5.0/lib/libexpat.so.1 (0x00007111e1dd0000)
libgcc_s.so.1 => /gnu/store/6ncav55lbk5kqvwwflrzcr41hp5jbq0c-gcc-11.3.0-lib/lib/libgcc_s.so.1 (0x00007111e1db6000)
libc.so.6 => /gnu/store/ln6hxqjvz6m9gdd9s97pivlqck7hzs99-glibc-2.35/lib/libc.so.6 (0x00007111e1bba000)
/gnu/store/ln6hxqjvz6m9gdd9s97pivlqck7hzs99-glibc-2.35/lib/ld-linux-x86-64.so.2 (0x00007111e22fb000)
```

- Cette version de python3 est liée avec la **glibc-2.35** (dans /gnu/store/ln6hxq...)
- La précédente version 3.8.2 était liée en **glibc-2.31**
- La glibc du système est **glibc-2.27** (mais n'est pas utilisée.)
- Le seul élément réellement utilisé dans le système hôte est le kernel (et ses drivers...)
- Très pratique pour continuer à disposer de logiciels à jour sur une plateforme ancienne (pour diverses raisons).

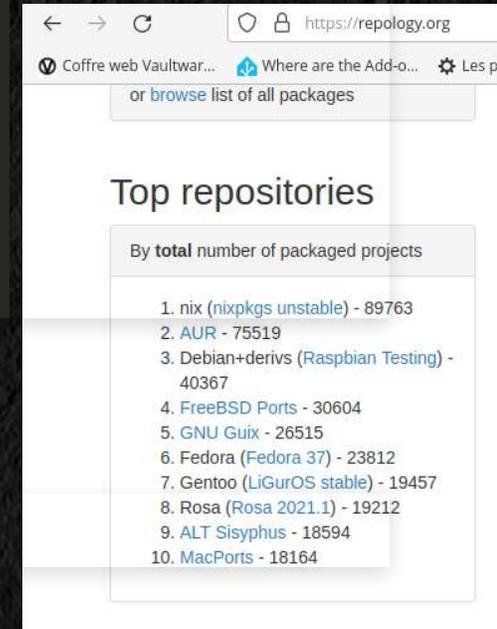
PAQUETS DISPONIBLES

```
guix package -A | grep gcc
[...]
gcc-toolchain 4.9.4 out,debug,static gnu/packages/commencement.scm:3551:4
gcc-toolchain 4.8.5 out,debug,static gnu/packages/commencement.scm:3551:4
gcc-toolchain 13.2.0 out,debug,static gnu/packages/commencement.scm:3551:4
gcc-toolchain 12.3.0 out,debug,static gnu/packages/commencement.scm:3551:4
gcc-toolchain 11.3.0 out,debug,static gnu/packages/commencement.scm:3551:4
gcc-toolchain 10.5.0 out,debug,static gnu/packages/commencement.scm:3551:4
[...]
```

- En terme de nombres de paquets disponibles :

```
moi@Scarab:~$ guix package -A
28796 115184 2770473
```

- Plus de 28.000 paquets de base. Le monde entier n'y est pas, mais les mainteneurs y travaillent. (5^e distribution Linux selon Repology)
- Des canaux supplémentaires existent
- Certains centres HPC en proposent



NAVIGATION ENTRE LES PROFILS

```
guix install gcc-toolchain@8
Le paquet suivant sera mis à une version inférieure :
  gcc-toolchain 13.2.0 → 8.5.0
[...]
moi@Scarab:~$ gcc --version
gcc (GCC) 8.5.0
```

```
guix package -l
[...]
Génération 3 26 janv. 2024 18:52:47 (actuelle)
+ gcc-toolchain 8.5.0 out /gnu/store/3d4ncn5z8qm56li86xfxr7jfybg9mmiy-gcc-toolchain-8.5.0
- gcc-toolchain 13.2.0 out /gnu/store/i3x1cr4b0f20r8pkcfq7sxmwwv6261b0y-gcc-toolchain-13.2.0
```

... Mon logiciel ne se compile plus ! Est-il possible de revenir en arrière ?

```
guix package --roll-back
passé de la génération 3 à 2
moi@Scarab:~$ guix package -l
[...]
Génération 2 26 janv. 2024 17:37:17 (actuelle)
python 3.10.7 out /gnu/store/8s8k2v6b5b7lyf5y5n534cjd9sipgas6-python-3.10.7
gcc-toolchain 13.2.0 out /gnu/store/i3x1cr4b0f20r8pkcfq7sxmwwv6261b0y-gcc-toolchain-13.2.0

Génération 3 26 janv. 2024 18:52:47
+ gcc-toolchain 8.5.0 out /gnu/store/3d4ncn5z8qm56li86xfxr7jfybg9mmiy-gcc-toolchain-8.5.0
- gcc-toolchain 13.2.0 out /gnu/store/i3x1cr4b0f20r8pkcfq7sxmwwv6261b0y-gcc-toolchain-13.2.0
```

Il est ainsi possible de revenir instantanément à d'anciennes installations !

LES RÉPERTOIRES DE GUIX

- /gnu/store contient tous les logiciels, mais aussi les profils. Il est volumétrique. Il est **READ-ONLY**, seul un démon peut y écrire.
 - Pas de logiciels `suid` autorisés.
- /var/guix contient essentiellement les générations de profils (des liens). Occupe peu de place.
- ~/.guix-profile pointe le profil courant de l'utilisateur (/var/guix/profiles/per-user/moi/guix-profile)
- ~/.config/guix/current pointe sur la génération de guix courante de l'utilisateur (/var/guix/profiles/per-user/moi/current-guix)
- Par défaut, le profil n'est pas chargé : la commande guix ne sera pas trouvée (ou seule la vieille du système). Les commandes installées ne sont pas trouvées. Il faut donc à **chaque login** faire :

```
source ~/.config/guix/current/etc/profile  
source ~/.guix-profile/etc/profile
```

Ces 2 commandes peuvent être systématisées dans le (.profile, .bashrc...) mais...



Toute commande ou librairie installée par guix deviendra alors préférentielle à celle du système. Cela peut être désirable ou non.

INSTALLER GUIX-DAEMON

Pour pouvoir écrire dans /gnu/store, il faut un démon qui tourne avec les droits root.

Sur des debian ou ubuntu récentes, apt-get install guix suffira !

Sinon, suivre l'installation binaire (documentation sur le site).

- Sinon, guix pack.
 - Permet de créer une image transportable d'un ou plusieurs logiciels
 - Au format tar, docker, singularity ...
 - Mais moins pratique qu'une commande guix native.

LE STORE

- Sa structure est /gnu/store/ <unhash> /lelogiciel.
- Guix est un gestionnaire de paquets fonctionnels. C'est une fonction qui prend en entrée les sources, les options de compilations, les dépendances logicielles... Et génère une sortie : le logiciel (avec un hash unique).
- Pour des paramètres d'entrées identiques (les mêmes sources, les mêmes options, les mêmes dépendances), le hash sera le même.
- 2 utilisateurs sur une machine installant un logiciel ayant le même hash **partageront** la même entrée du /gnu/store !
- Dans une installation multi-utilisateurs, multi-machines, ce store est partageable (clusters de calcul)

CONSTRUCTION DEPUIS LES SOURCES

- Les logiciels sont entièrement construits depuis les sources.
- Par défaut, ils sont compilés sur la machine.
 - Ce qui prend un temps considérable !
 - Sauf si des serveurs de substitut ont été configurés
 - (C'est le cas par défaut)
 - Si le même hash existe, c'est le même logiciel... il est téléchargé précompilé
- La compilation peut être déchargée sur une machine plus puissante
- Tout paquet original (provenant de canaux non officiels ou d'une dérivation personnelle) sera recompilé à chaque modification d'une de ses dépendances (pas de substitut disponible).
 - À moins d'avoir un serveur (cui-rass) qui le fait automatiquement.

PLUSIEURS VERSIONS EN MÊME TEMPS

(Partie Nicolas Vallet)

CONCLUSIONS

- Reprenons les points :

« JE N'EN AI PAS L'USAGE »

- Installer des logiciels...
 - À jour sur une machine plus ancienne
 - Non disponibles ailleurs
 - Sans être root
 - Sans interférence
 - Revenir en arrière à tout moment
- Modifier à l'infini ces logiciels (plus compliqué)
- Les optimiser
- Créer à la volée des environnements ad-hoc

« GUIX EST TROP COMPLIQUÉ »

- guix install
- guix remove
- guix pull
 - Attention, vrai point bloquant : cette opération est longue.
- guix upgrade

Pas vraiment plus compliqué. Ça peut le devenir si l'utilisateur veut faire plus... (mais c'est aussi le cas ailleurs)

« DOCKER/APPTAINER/CONDA FONT PAREIL »

- Récupération de **binaires** et de toutes les dépendances. Où est la confiance ?
 - Il y a eu de nombreuses attaques relatives à ces soucis
- Les « strates » ne sont pas souvent partagées (apptainer/conda)
- L'opacité des solutions déployées et leur non-évolutivité
- guix shell permet de faire l'équivalent lorsque c'est nécessaire

« IL N'Y A PAS BEAUCOUP DE LOGICIELS »

28.000 avec augmentation régulière (plus ceux fournis par des canaux annexes)

« ÇA N'EST PAS AUSSI BIEN QUE CE QUE JE COMPILE À LA MAIN »»

- Pas de conteneurisation par défaut
- Pas de couche filesystem supplémentaire type overlaysfs
- Performances natives
- Possibilité de tuning (les optimisations par défaut peuvent être conservatrices)
- Facile à customiser (mais nécessite quelques connaissances)
- Pile complète de dépendances optimisable
- Suivi des mises à jour



C'est l'utilisateur qui a le pouvoir !



2

4

3

15

14

5

13

MERCI DE VOTRE ATTENTION

Questions ?

12

6

11

7