

# JDEV2017 Atelier T8.AP04

## NIX & HPC: prise en main

GRICAD - Pôle Calcul

July 4, 2017



GRENOBLE ALPES RECHERCHE  
INFRASTRUCTURE DE  
CALCUL INTENSIF  
ET DE DONNÉES





## Atelier T8.AP04

### ▶ Part. I) Parlons Nix

- \* Contexte
- \* Nix - Vue d'ensemble et concepts
- \* Nix Advanced - Concepts avancés
- \* Les liens utiles

### ▶ Part. II) Prise en main (TP)

### ▶ Part. III) Installation de Nix sur un ordinateur HPC



## CONTEXTE

## Les "modules" d'environnement

- ▶ Mettre des bibliothèques et des applications à la disposition des utilisateurs
- ▶ Chaque application a des dépendances précises: librairie + version + compilateur + options de compilation + ...
- ▶ La commande "**module**" est largement utilisée par les **centres HPC** pour répondre à ces besoins

```
module load intel-compiler2017
```

## Les "modules" d'environnement

- ▶ Mettre des bibliothèques et des applications à la disposition des utilisateurs
- ▶ Chaque application a des dépendances précises: librairie + version + compilateur + options de compilation + ...
- ▶ La commande "**module**" est largement utilisée par les **centres HPC** pour répondre à ces besoins

```
module load intel-compiler2017
```

### Problèmes:

- ▶ Difficile à maintenir: multiplication des dépendances
- ▶ Cohérence du dépôt d'appli uniquement maintenu par `root`
- ▶ Reproductibilité ?
- ▶ Portabilité ?

⇒ **Besoin** d'une solution plus **efficace** et que l'on pourrait **partager**!

## Les containers: une solution ?



- ▶ Configuration des containers (sécurité, réseau, ...)
- ▶ Maintenance des images
- ▶ Reproductibilité?

## Easy build



*User level automatic building*

<https://hpcugent.github.io/easybuild/>

## Nix



*A packaging system that allows user-level installs*

<https://nixos.org/nix/>

## Spack



# Spack

*User level automatic building*

<https://github.com/LLNL/spack>

## Guix



<https://www.gnu.org/software/guix>

<https://www.gnu.org/software/guix>



## NIX - VUE D'ENSEMBLE ET CONCEPTS





## Nix

- ▶ Functional package manager
- ▶ Fiable & reproductible
- ▶ Dispo sur Linux & MAC OS
- ▶ Users peuvent créer/installer un paquet sans passer `root`

## Nixpkgs

- ▶ 6,500 paquets
- ▶ Pur: pas de deps en dehors du "Nix store"

## NixOS

The Purely Functional Linux Distribution



## Nix

- ▶ Functional package manager → Pas d'effet de bord
- ▶ Fiable & reproductible → Expérimentation, recherche
- ▶ Dispo sur Linux & MAC OS → Un même paquet pour Tier0/1/2/3
- ▶ Users peuvent créer/installer un paquet sans passer root → Facilité de la personnalisation d'env.

## Nixpkgs

- ▶ 6,500 paquets → Partage
- ▶ Pur: pas de deps en dehors du "Nix store"

## NixOS

The Purely Functional Linux Distribution → Pour aller plus loin



## Stockage des paquets

**Unique répertoire** `/nix/store`

→ **Pas de pollution de l'arborescence du système**



## Stockage des paquets

**Unique répertoire** `/nix/store`

→ **Pas de pollution de l'arborescence du système**

## Identification des paquets via hash

**Construction du paquet** → **unique sous-répertoire**

`/nix/store/an9dli66ng2jzvqf13b2i230mm9fq7qk-cdo-1.7.2`

*Hash = mix(sources + deps + flags, ...)*

Compilation + nouvelle option → nouveau hash

`/nix/store/srf6grrfy9vkc9fsplk8xk292lm8jvz5-cdo-1.7.2`

→ **Conservation de l'arbre des dépendances,**

→ **Unicité du paquet**

## EN RÉSUMÉ

- ▶ Installation d'un paquet = Création de liens dans le profil de l'utilisateur stocké dans son home
- ▶ `PATH=~/.nix-profile/bin:$PATH`

```
~$ nix-env -i hello

~$ readlink 'which hello'
/nix/store/3dlqv87hrrfjynj0brbn4h71g4g4g89z-hello-2.10/bin/hello

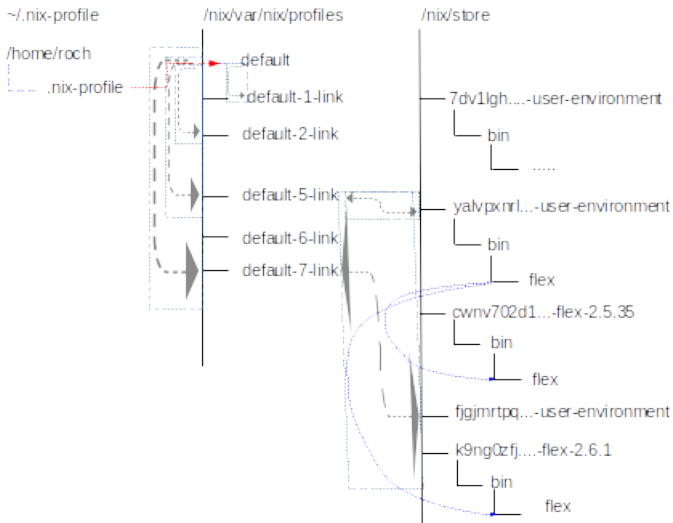
~$ ldd /nix/store/3dlqv87hrrfjynj0brbn4h71g4g4g89z-hello-2.10/bin/hello
        linux-vdso.so.1 (0x00007ffe5d1b6000)
        libc.so.6 =>
/nix/store/q3wx1gab2ysnk5nyvvyg56ana2v4r2ar-glibc-2.24/lib/libc.so.6
(0x00007f3d90bc9000)
/nix/store/q3wx1gab2ysnk5nyvvyg56ana2v4r2ar-glibc-2.24/lib/ld-linux-x86-64.so.2
(0x00007f3d90f67000)
```



## NIX ADVANCED - CONCEPTS AVANCÉS

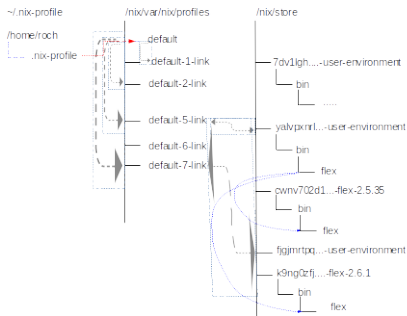
# Nix - Les concepts

## Profile



## Profile

Environnement défini par un **ensemble de liens symboliques** dans le répertoire personnel de l'utilisateur, vers les paquets utilisés dans le store.



- ▶ **Nombre illimité et cohabitation** de profiles
- ▶ **Suppression, mise à jour du profile**  
=> aucun effet pour le reste
- ▶ **Rollbacks** et historique des générations





### channel

tgz du snapshot de Nixpkgs

Ex. de channels:

- \* `nixpkgs-unstable`
- \* `nixos-YY.MM` (NixOS-users)
- \* `ciment-channel` (CIMENT users :)

Exemple: Installation du paquet `openmpi`

```
$ nix-env -i -A nixpkgs-unstable.openmpi
```

```
$ nix-env -i -A ciment-channel.openmpi
```



## channel

tgz du snapshot de Nixpkgs

Ex. de channels:

- \* `nixpkgs-unstable`
- \* `nixos-YY.MM` (NixOS-users)
- \* `ciment-channel` (CIMENT users :)

Exemple: Installation du paquet `openmpi`

```
$ nix-env -i -A nixpkgs-unstable.openmpi
```

```
$ nix-env -i -A ciment-channel.openmpi
```

## binary cache

Pour installation d'un paquet:

1. Parcours du `binary-cache`
2. Si besoin, construction depuis les sources et des deps

## Pour faire court

- ▶ Langage fonctionnel (*Dérivé d'Haskell*) → pas d'effet de bord
- ▶ Pas d'affectation de variable → pur
- ▶ Argument de fonction utilisé **au besoin** → évaluation paresseuse
- ▶ Types: entiers, opérateurs, caractères, listes, ensembles, ...

Essayer le langage avec la commande `nix-repl`:

```
$ nix-env -i nix-repl
$ nix-repl
nix-repl> 1+3
4
nix-repl> builtins.div 6 3
2
```

source: <http://lethalman.blogspot.fr/2014/07/nix-pill-4-basics-of-language.html>

### Installation d'une variante de gromacs

```
$ nox gromacs  
Refreshing cache  
1 gromacs-4.6.7 (ciment-channel.gromacs )  
Molecular dynamics software package  
2 gromacs-4.6.7 (ciment-channel.gromacsDouble )  
Molecular dynamics software package  
3 gromacs-4.6.7 (ciment-channel.gromacsDoubleMpi )  
Molecular dynamics software package  
4 gromacs-4.6.7 (ciment-channel.gromacsMpi )  
Molecular dynamics software package
```

Légende: attribut

### Dérivation

fonction d'action de construction du langage qui prend en paramètre des arguments de la construction

```
{ stdenv, fetchurl, cmake,
  singlePrec ? true,
  mpiEnabled ? false,
  fftw,
  openmpi
}:
stdenv.mkDerivation {
  name = "gromacs-4.6.7";

  src = fetchurl {
    url = "ftp://ftp.gromacs.org/pub/gromacs/gromacs-4.6.7.tar.gz";
    sha256 = "6afb1837e363192043de34b188ca3cf83db6bd189601f2001a1fc5b0b2a214d9";
  };
  buildInputs = [cmake fftw];
  ++ (stdenv.lib.optionals mpiEnabled [ openmpi ]);

  cmakeFlags = ''
    ${if singlePrec then "-DGMX_DOUBLE=OFF" else "-DGMX_DOUBLE=ON -DGMX_DEFAULT_SUFFIX=OFF"}
    ${if mpiEnabled then "-DGMX_MPI:BOOL=TRUE
      -DGMX_CPU_ACCELERATION:STRING=SSE4.1
      -DGMX_OPENMP:BOOL=TRUE
      -DGMX_THREAD_MPI:BOOL=FALSE"
      else "-DGMX_MPI:BOOL=FALSE" }
  '';
  meta = with stdenv.lib; {
    homepage = "http://www.gromacs.org";
    license = licenses.gpl2;
    description = "Molecular dynamics software package";
    platforms = platforms.unix;
  };
};
```



## REFS & BIBLIO

## Biblio officielle

**Nix:** <https://nixos.org/nix/>

**NixOS:** <https://nixos.org/>

**Nixpkgs:** <https://nixos.org/nixpkgs/>

## Quelques références GRICAD

**Blog:** <https://gricad.github.io/calcul/>

**Channel:** <https://github.com/Gricad/nix-ciment-channel>

## Autres

**Blog:** <http://lethalman.blogspot.fr/2014/07/nix-pill-1-why-you-should-give-it-try.html>

## JDEV2017

**Atelier Préparatoire:** T8.AP04

**Groupe de Travail :** T8.GT06

**Poster:** Contributing to the Nix HPC packages collection



**Merci !**

N'hésitez pas à poser vos questions ...

