

Guix vs Conda : gestion d'environnements logiciels pour un projet scientifique

Konrad Hinsén

Centre de Biophysique Moléculaire, Orléans, France
Synchrotron SOLEIL, Saint Aubin, France

31 mars 2026

Vous êtes ici: Accueil > Cours > Reproducible Research II: Practices and tools for managing computations and data

Recherche de cours

Outils pour la recherche

Numérique et technologie

Reproducible Research II: Practices and tools for managing computations and data

Réf. 41023

In this MOOC, we will show you how to improve your practices and your ability to manage and process larger amounts of data, complex computations, while controlling your software environment.

🕒 Effort : 35 heures

🕒 Rythme: Auto-rythmé

🌐 Langues: Anglais



Aucune session ouverte



C'est quoi un environnement logiciel ?

Tout ce qu'il faut pour exécuter un programme.

Tout ce qui peut influencer le résultat d'un programme.

- Le matériel : processeur, mémoire, périphériques, ...
- Les fichiers accessibles (dont les logiciels)
- Les ressources accessibles par réseau
- Les variables d'environnement

Exemple demo.py

```
import platform
import subprocess

result1 = platform.platform()
print("Result_1:")
print(result1)

result2 = open("/etc/motd").read()
print("Result_2:")
print(result2)

result3 = subprocess.run(["ls", "/"], stdout=subprocess.PIPE)
print("Result_3:")
print(result3.stdout.decode('utf-8'))
```

Objectif reproductibilité

Obtenir les mêmes résultats :

- un an plus tard (ou deux, ou dix, ou 50, ...)
- sur un autre ordinateur
- par une autre personne

Mission impossible :

- Le matériel a une durée de vie limitée
- Les ressources réseau sont hors contrôle

Compromis :

- Définir des priorités
- Identifier les éléments les plus problématiques
- Accepter des contraintes imposées aux programmes

Idéalement :

- Limiter l'accès aux fichiers autant que possible
→ conteneur
- Limiter l'accès aux ressources par réseau
→ pare-feu
- Maîtriser l'évolution des logiciels
→ gestionnaire de paquets
→ sans passer par un administrateur

Conteneurs

- Guix : `guix shell -C ...`
- Conda : délègue à Docker

Pare-feu

- Guix : `guix shell -N ...` (tout ou rien)
- Conda : délègue à Docker

Logiciels

- Guix : environnement déclaratif
- Conda : environnement composé par installation de paquets

Environnements pour la recherche

Guix

- Deux fichiers (versionnés, partagés, publiés, ...) :
 - liste des paquets : `manifest.scm`
 - versions : `channels.scm`
- `guix time-machine -C channels.scm -- shell -m manifest.scm`

Conda

- `conda create -n mon-env paquet1=version1 paquet2 ...`
- `conda install paquet3=version3 ...`
- `conda activate mon-env`
- calculs...
- `conda export > mon-env.yml`
- `conda create --file mon-env.yml`

Conda à long terme

Tentons un environnement créé en 2023 :

```
$ mamba create --file mooc-env.yml
```

bioconda/noarch	5.5MB @	1.5MB/s	3.5s
bioconda/linux-64	5.5MB @	1.4MB/s	3.8s
conda-forge/noarch	25.4MB @	3.1MB/s	8.2s
conda-forge/linux-64	52.2MB @	3.9MB/s	13.7s

```
error libmamba Could not solve for environment specs
```

```
  The following packages are incompatible
```

```
  |-- zstandard =0.19.0 py311hbe0fcd7_1 is installable and it requires
```

```
  | |-- zstd ==1.5.2 *, which can be installed;
```

```
  |-- zstd =1.5.5 hfc55251_0 is not installable because it conflicts with any installable v
```

```
critical libmamba Could not solve for environment specs
```

Faux remèdes :

- `conda export -no-build > mooc-env.yml`
- `conda export -from-history > mooc-env.yml`

Plus d'erreur, mais on obtient un environnement différent !

Guix vs. Conda : les priorités

Guix

- Reproductibilité

Conda

- Facilité de déploiement multi-plateforme (Linux, macOS, Windows)
- Facilité d'évolution par spécification partielle (conda create -n test python=3.9 numpy)
- Images Docker compactes

Comment rendre un calcul reproductible ?

Input

```
100111100001001100110101101100
001010011101010111110001001101
010111101100011110111011110001
001100001110111000100100000111
110101100111001110100000100110
110111100111000011111101101111
111001001011110001100110000101
011100001000010001011110000010
110101110011101111001010100111
111000101110011001101101001001
011001010100101011000001001100
11010011100101111100001011101
01111011111000111011110101101
000001110110011001010101011100
100010110001100000111001100010
000000111011100100100101010111
000010000001100001000010110110
101111101111000111100101110101
100101010100001001110100010001
011110011010100101111011110101
100011000110110001011101100110
110100000100000011011000001101
100000011100100111101101011011
010110010001000101110111001010
```



Output

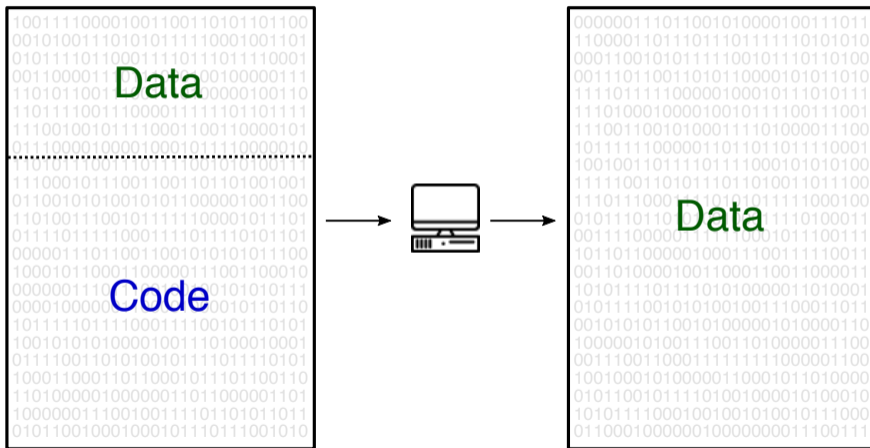
```
000000111011001010000100111011
110000110111011101111110101010
000110010101111100101110110100
001110110011010110000101011010
111101111100000100010111010111
111010001000010010111100111001
111001100101000111101000011100
101111110000011011011011110001
100100110111101111000101010100
111110011010111011010011011100
111011100011110101011111000100
010111011010100100011110100011
00111100000111110001011100111
101101100000100011100111110011
001101000010011000110011000011
101011110111101010000011010001
010111100101010010011100011011
001010101100101000001010000110
100000101001110011010000011100
0011100110001111111111000001100
100100010100000110001011010000
010110010111101001000010100010
101011110001001001010010111000
01100010000001000000011100111
```

Computer by Creative Stall from the Noun Project

Comment rendre un calcul reproductible ?

Input

Output

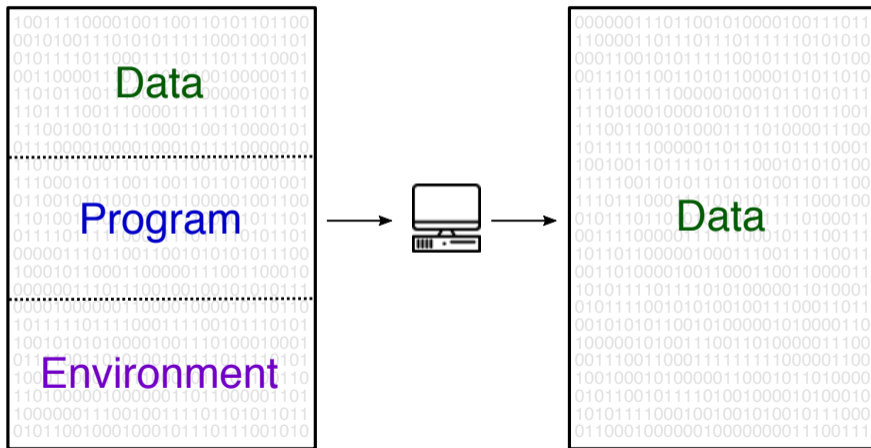


Computer by Creative Stall | from the Noun Project

Comment rendre un calcul reproductible ?

Input

Output

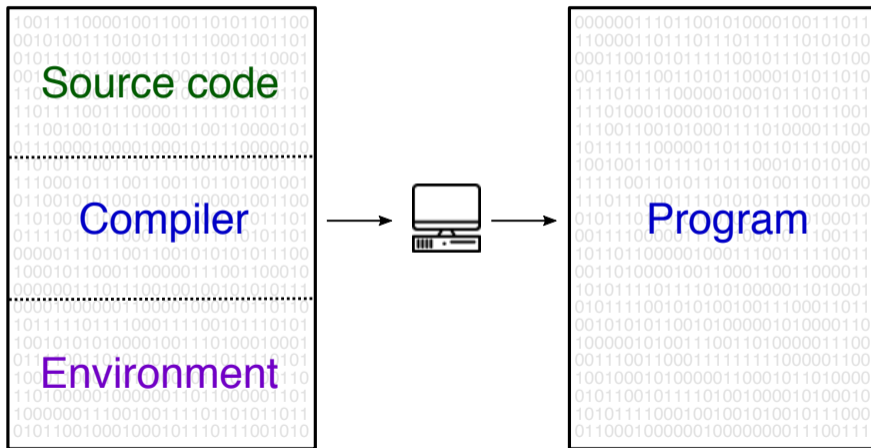


Computer by Creative Stall from the Noun Project

La compilation est un calcul

Input

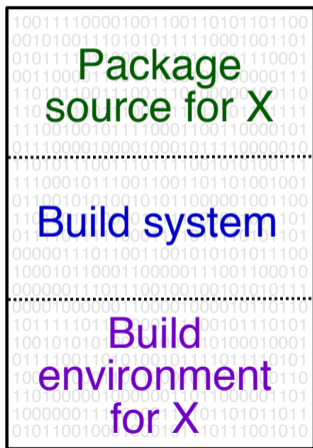
Output



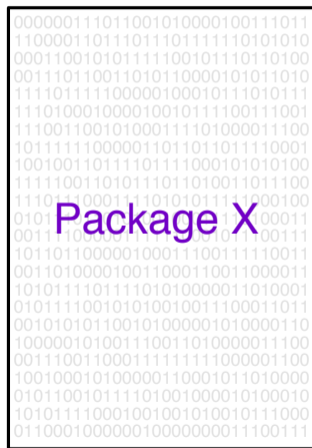
Computer by Creative Stall from the Noun Project

La construction d'un paquet est un calcul

Input

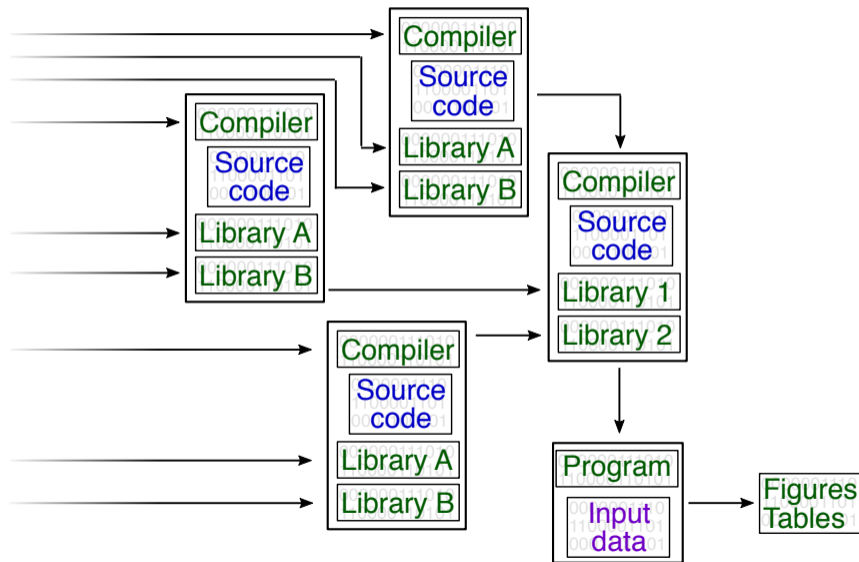


Output



Computer by Creative Stall from the Noun Project

Les étapes d'un calcul



Comment rendre un calcul reproductible ?

Rendre toutes ses étapes reproductibles :

- Mêmes données d'entrée (y compris code source)
- Mêmes exécutables
- Même environnement

Les exécutables sont des résultats de calcul !

→ reproductibilité récursive

Ça marche pareil avec NumPy 2.1 et matplotlib 3.9 ?

Explosion combinatoire

Approche conda :

- contraintes de version moins strictes
- satisfaction des contraintes par un solveur

Résultat du solveur non reproductible !

Interfaçage avec le système d'exploitation

Guix

- Noyau Linux

Très stable

Conda

- glibc sous Linux
- libSystem etc. sous macOS
- MSVC runtime sous Windows

Images Docker sans glibc

Facilite le déploiement multi-plateforme.

Pas de reproductibilité en cas de changement de libc !

Conclusions

- Deux gestionnaires d'environnement avec des priorités différentes
- Choisissez en fonction de vos propres priorités
- La doc de conda parle beaucoup de reproductibilité, mais sans la définir précisément
- En pratique : reproductibilité de courte durée, sur la même plate-forme

Caricature :

- **Guix : galère au début, facilité après**
- **Conda : facilité au début, galère après**

Outils pour la recherche

Numérique et technologie

Reproducible Research II: Practices and tools for managing computations and data

Réf. 41023

In this MOOC, we will show you how to improve your practices and your ability to manage and process larger amounts of data, complex computations, while controlling your software environment.

🕒 Effort : 35 heures

🕒 Rythme: Auto-rythmé

🌐 Langues: Anglais



Aucune session ouverte

