

Un exemple d'utilisation de Guix pour gérer son environnement de développement

Exemple avec le projet MSE mse.biosp.org



Olivier Bonnefon

décembre 2025

Plan

- 1 Difficultés rencontrées à la construction de l'environnement de développement
- 2 Intérêt à utiliser Guix
- 3 Exemples de recettes
- 4 Canaux Guix
- 5 Résumé de la démarche

Contenu d'un environnement de développement

- des outils algorithmiques, numériques : dolfinx/petsc/gmsh/scotch/hdf5/...
 - des outils de compilation: automake/cmake@4.03/gcc/gcc-toolchain/python-setuptools...
 - des outils de dev: emacs/ddd/valgrind/....
- guix graph env-to-compil-mse : 1830 paquets
guix graph python-mse : 1010 paquets

Contenu d'un environnement de développement

- des outils algorithmiques, numériques : dolfinx/petsc/gmsh/scotch/hdf5/...
- des outils de compilation: automake/cmake@4.03/gcc/gcc-toolchain/python-setuptools...
- des outils de dev: emacs/ddd/valgrind/....
guix graph env-to-compil-mse : 1830 paquets
guix graph python-mse : 1010 paquets

Conclusion

Il est plus difficile de construire l'environnement de développement que l'environnement d'utilisation de MSE.

Difficultés liées à la construction de l'environnement

- incompatibilité de version de bibliothèques partagées (ex: hdf5) \implies nécessité de compiler les outils numériques
- compilation des outils utilisés: procédures souvent hasardeuses et incomplètes
- difficulté à reproduire ces configurations sur d'autres machines
- subir les saut de versions des outils utilisés

Difficultés liées à la construction de l'environnement

- incompatibilité de version de bibliothèques partagées (ex: hdf5) \implies nécessité de compiler les outils numériques
- compilation des outils utilisés: procédures souvent hasardeuses et incomplètes
- difficulté à reproduire ces configurations sur d'autres machines
- subir les saut de versions des outils utilisés

Sources de stress:

- lors de l'accueil d'un nouveau collaborateur
- lors d'un changement de machine
- lors d'allers retours entre des projets
- lors de sauts de versions subis (ex: Package 'rgdal' was removed from the CRAN repository)

Difficultés liées à la construction de l'environnement

- incompatibilité de version de bibliothèques partagées (ex: hdf5) \implies nécessité de compiler les outils numériques
- compilation des outils utilisés: procédures souvent hasardeuses et incomplètes
- difficulté à reproduire ces configurations sur d'autres machines
- subir les saut de versions des outils utilisés

Sources de stress:

- lors de l'accueil d'un nouveau collaborateur
- lors d'un changement de machine
- lors d'allers retours entre des projets
- lors de sauts de versions subis (ex: Package 'rgdal' was removed from the CRAN repository)

Diagnostique: Manque de reproductibilité des environnements de développement

Intérêt à utiliser Guix

Intérêt à utiliser Guix

Guix peut apporter la reproductibilité

- dans l'espace: d'une machine à l'autre en reproduisant le même environnement binaire d'une machine à l'autre
- dans le temps: étiquetage de sources et usage des canaux.

Intérêt à utiliser Guix

Guix peut apporter la reproductibilité

- dans l'espace: d'une machine à l'autre en reproduisant le même environnement binaire d'une machine à l'autre
- dans le temps: étiquetage de sources et usage des canaux.

Il n'y a pas de magie, il faut:

- mettre au point les recettes Guix pour gérer les conflits de versions
- mettre à jour les recettes lorsque je choisis de faire un saut de version

Exemples de recettes

recette pour les développeurs "env-to-compil-mse"

```
(define-public env-to-compil-mse
  (package
    (name "env-to-compil-mse")
    ...
    (build-system trivial-build-system)
    (propagated-inputs
      (list python-setuptools python-wheel python-sphinx
            vim grep cmake gcc-toolchain
            dolfinx gmsh petsc openmpi ... ))
    (synopsis "Development environment for MSE")
  ))
```

recette versionnée sur forge.inrae.fr/msegrp/guix-biosp

recette pour les utilisateurs "python-mse"

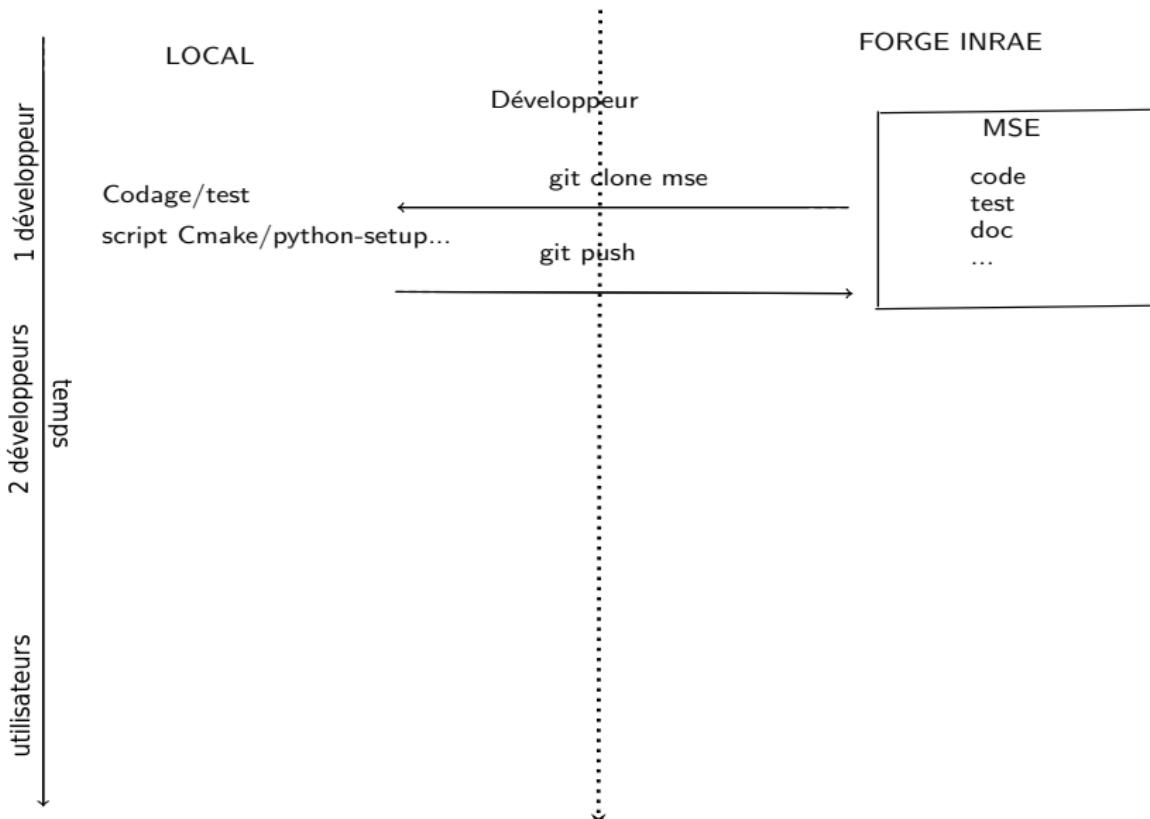
```
(define-public python-mse
  (package
    (name "python-mse")
    (version "v2.4.4")
    (source
      (uri (string-append
              "https://forgemia.inra.fr/msegrp/mse/-/
              archive/" version "/mse-" version ".tar.gz")
            (sha256(base32
"08i2j4sp5p33wxgcfjd2clj1a1mcwyvlf19n8d19lq4x280a33zf"))))
    (build-system cmake-build-system)
    (arguments ...)
    (native-inputs (list cmake swig python-wheel...))
    (propagated-inputs (list dolinx petsc gmsh openmpi ...))
    (synopsis "environnement méca-stat")
    (description "simulation et estimation d'EDP")
  )
```

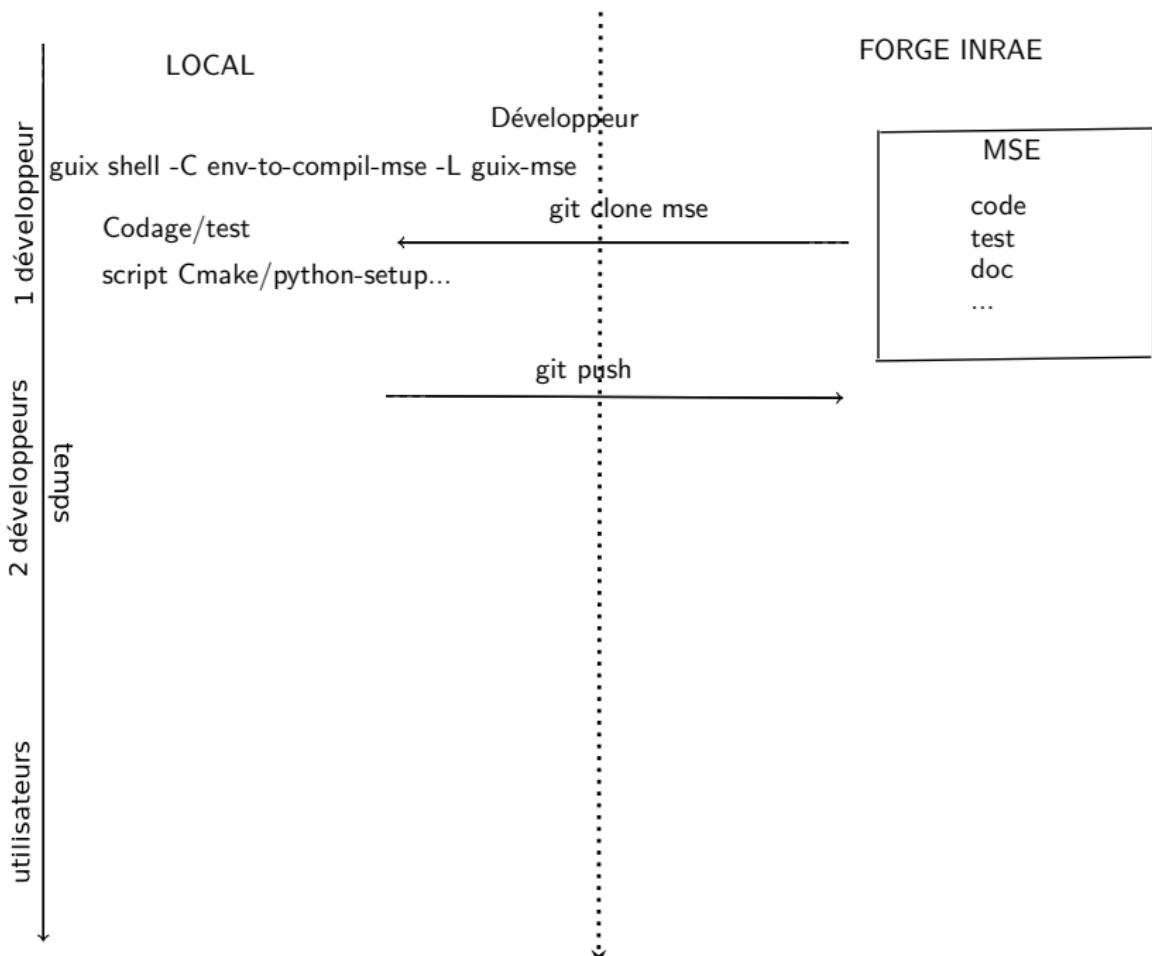
Partage et reproductibilité: canaux Guix

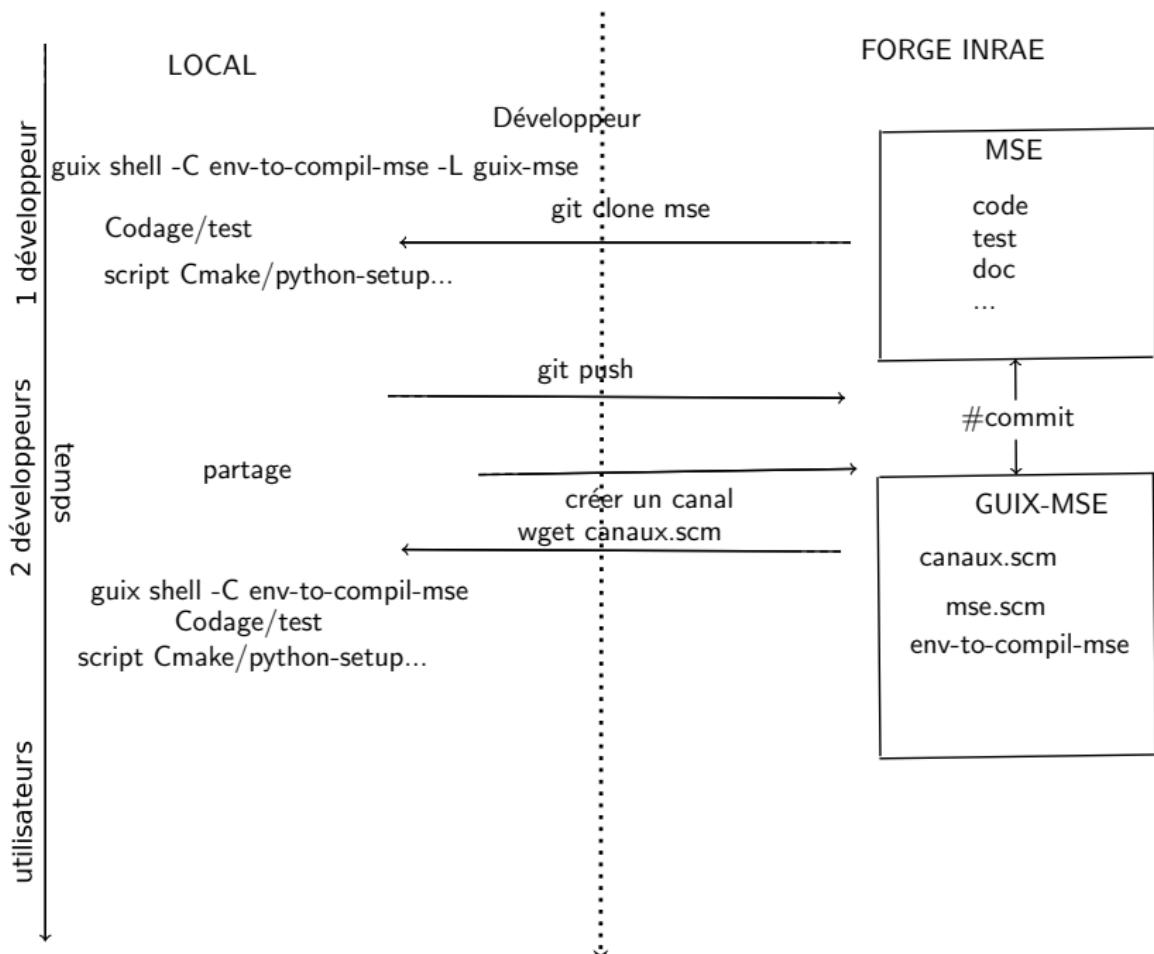
```
$ cat ~/.config/guix/channels.scm
(list (channel (name 'guix)
  (url "https://codeberg.org/guix/guix")
  (branch "master")
  (commit "adde76d065a053a0ef7cb529edccc1f7f0467b35"))
  (channel (name 'guix-science)
    (url "https://codeberg.org/guix-science/guix-science")
    (branch "master")
    (commit "dbdbbeb15b63e49ebb1af0fcc109e892e04b4748e"))
  (channel (name 'guix-biosp)
    (url "https://forge.inrae.fr/msegrp/guix-biosp.git")
    (branch "pre-v2.5")
    (commit "0f045eaf7b753e88f8b45754f8e0857647bd4560"))))
```

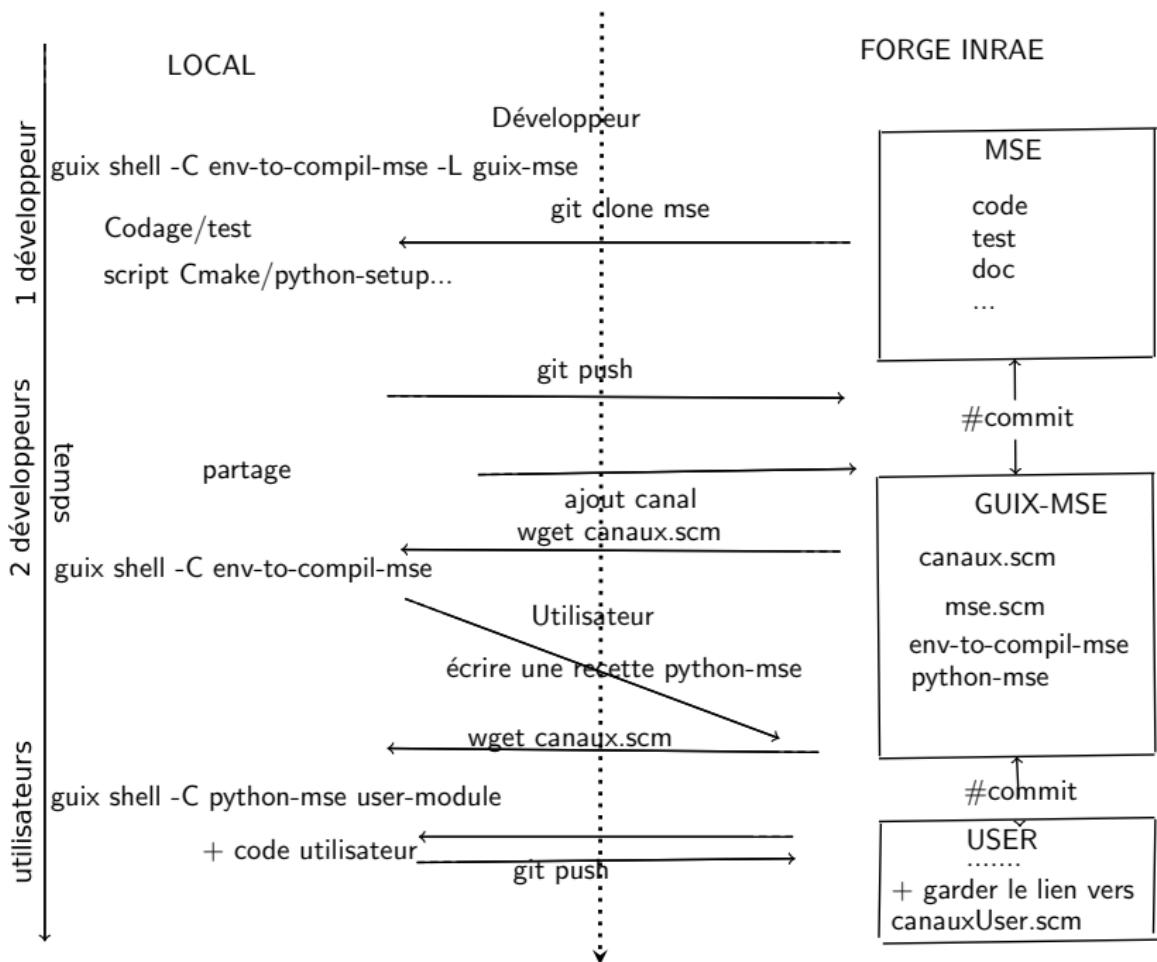
- Ce fichier channels.scm pour MSE est archivé sur forge.inrae.
- L'objectif est maintenant de l'intégrer à guix-science.

Résumé de la démarche









Mise en œuvre sur le cluster de BioSp.

- Guix: <https://guix.gnu.org/>
- café Guix: <https://hpc.guix.info/events/2024-2025/caf%C3%A9-guix/>

Merci pour votre attention.

Questions / remarques?