



guix time-machine **et** guix describe

Café Guix

28 mai 2024

Marek Felšöci
marek.felsoci@inria.fr

Parcours universitaire

- Licence, Master en informatique (Université de Strasbourg)
- Doctorat en informatique (Université de Bordeaux)

Thématiques de travail

- calcul haute-performance
 - algorithmique d'algèbre linéaire
- compilation source-à-source
 - optimisation de boucles, parallélisation automatique
- méthodes de recherche reproductible

guix time-machine [9] et guix describe [6]

Plan

- mise en situation
- quelques rappels de Guix [4, 11]
- voyage dans le temps
- utilisation plus avancée
- exemples de mise en pratique



<https://tinyurl.com/cafe-guix>

Reprise d'un projet précédent

Printemps 2022

- développement d'une application d'algèbre linéaire
- environnement Guix avec GCC, MUMPS [1, 2], shell de base

```
guix shell --container gcc-toolchain mumps bash coreutils -- bash
```

Reprise d'un projet précédent

Printemps 2022

- développement d'une application d'algèbre linéaire
- environnement Guix avec GCC, MUMPS [1, 2], shell de base

```
guix shell --container gcc-toolchain mumps bash coreutils -- bash
```

De nos jours

- reprise du projet → la même ligne de commande `guix shell [8]`

Questions

Obtiendrai-je exactement le même environnement ?

Questions

Obtiendrai-je exactement le même environnement ?

- MUMPS 5.2.1 → MUMPS 5.5.1
- GCC 11.2.0 → GCC 13.2.0

Questions

Obtiendrai-je exactement le même environnement ?

- MUMPS 5.2.1 → MUMPS 5.5.1
- GCC 11.2.0 → GCC 13.2.0

```
guix shell --container gcc-toolchain@11.2.0 mumps@5.2.1 bash  
↪ coreutils -- bash
```

- MUMPS 5.2.1 n'est plus disponible
- et les autres dépendances?

Questions

Obtiendrai-je exactement le même environnement ?

- MUMPS 5.2.1 → MUMPS 5.5.1
- GCC 11.2.0 → GCC 13.2.0

```
guix shell --container gcc-toolchain@11.2.0 mumps@5.2.1 bash  
↪ coreutils -- bash
```

- MUMPS 5.2.1 n'est plus disponible
- et les autres dépendances?

Que me faudrait-il pour retrouver mon environnement ?

Questions

Obtiendrais-tu exactement le même environnement ?

- MUMPS 5.2.1 → MUMPS 5.5.1
- GCC 11.2.0 → GCC 13.2.0

```
guix shell --container gcc-toolchain@11.2.0 mumps@5.2.1 bash  
↪ coreutils -- bash
```

- MUMPS 5.2.1 n'est plus disponible
- et les autres dépendances?

Que me faudrait-il pour retrouver mon environnement ?

- ah, une machine à voyager dans le temps devrait faire affaire
 - `guix time-machine` et `guix describe`

guix pull [7]

Canaux [3]

- mise à disposition de paquets, Guix y compris
 - canal officiel de Guix (28 026 paquets)
 - nombreux autres canaux
- dépôts git
 - tout changement est traçable à travers les *commits*

Age	Commit message (Expand)	Author
97 min.	gnu: emacs-are: Update to 0.9.5. HEAD master	Andrew Tropin
97 min.	gnu: guile-ares-rs: Update to 0.9.5.	Andrew Tropin
116 min.	gnu: cl-dissect: Update to 1.0.0-2.a70cab6.	Andre A. Gomes
2 hours	gnu: Add cl-data-structures.	Guillaume Le Vaillant
2 hours	gnu: Add cl-more-conditions.	Guillaume Le Vaillant
7 hours	tests: install: Use a default qemu machine for arm architectures.	Efrain Flashner
7 hours	gnu: ovmf: Deprecate in favor of ovmf-x86-64.	Efrain Flashner
7 hours	tests: install: Add support for aarch64 UEFI tests.	Efrain Flashner
7 hours	gnu: Use new ovmf firmware packages.	Efrain Flashner

- mise à jour ≈ télécharger les derniers *commits*
 - guix pull → nouvelle *génération*

guix shell

Usage

- crée des environnements logiciels sans changer mon profil actuel
- prends les versions par défaut de la génération courante

```
guix shell --container gcc-toolchain mumps bash coreutils -- \
  bash -c \
  'cat $GUIX_ENVIRONMENT/include/zmumps_c.h | head -n 5) &&
  ↪ gcc --version'
```

Printemps 2022

```
\* This file is part of MUMPS 5.2.1, released
gcc (GCC) 11.2.0
```

De nos jours

```
\* This file is part of MUMPS 5.5.1, released
gcc (GCC) 13.2.0
```

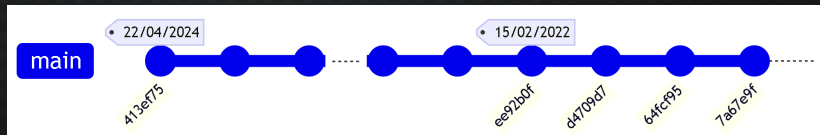
guix shell

Printemps 2022

```
\* This file is part of MUMPS 5.2.1, released  
gcc (GCC) 11.2.0
```

De nos jours

```
\* This file is part of MUMPS 5.5.1, released  
gcc (GCC) 13.2.0
```



Extrait du graphe de révisions du canal officiel de Guix.

Que me faut-il pour retrouver mon environnement ?

- remonter le temps et exécuter `guix shell` comme si c'était le printemps 2022
 - avec les canaux qui sont des dépôts git, c'est possible

```
guix time-machine --no-channel-files --commit=<bon-commit>  
↪ -- <commande guix>
```

Que me faut-il pour retrouver mon environnement ?

- remonter le temps et exécuter `guix shell` comme si c'était le printemps 2022
 - avec les canaux qui sont des dépôts git, c'est possible

```
guix time-machine --no-channel-files --commit=<bon-commit>  
↪ -- <commande guix>
```

Penser à garder la traces des canaux et *commits* utilisés !

guix describe

- afficher les informations sur la dernière génération de Guix dans notre profil

Pokolenie 1 22. máj 2024 02:30:17 (súčasn )

...

guix 413ef75

zdroj repozit ra: <https://git.savannah.gnu.org/git/guix.git>

vetva: master

 prava: 413ef75f89ac337f7ac3331a862c7c8cdc88aa64

- 413ef75 (22/05/2024) > ee92b0f (15/02/2022)

guix time-machine

- *commit* du canal + ligne de commande `guix shell`
 - → mon environnement d'origine

```
guix time-machine --no-channel-files \  
  --commit=ee92b0fb7dfd8b55ea63254a59a1605fb870a44c -- \  
  shell --container gcc-toolchain mumps bash coreutils -- \  
  ↪ bash
```

Résumé

guix time-machine

- exécuter une commande Guix dans une version antérieure de Guix
- utilisable avec `guix shell` mais aussi avec d'autres commandes
- attention au sens du mot « antérieure »

Résumé

guix time-machine

- exécuter une commande Guix dans une version antérieure de Guix
- utilisable avec `guix shell` mais aussi avec d'autres commandes
- attention au sens du mot « antérieure »

guix describe

- obtenir des informations sur éléments d'une génération de Guix
- numéros de *commit* au lieu des dates et heures

Cas d'utilisation

guix time-machine et guix shell

- 1 reprendre un travail précédent (vu ici)
- 2 naviguer dans un environnement en évolution
- 3 assurer un environnement stable

Pensez-vous à d'autres cas d'utilisation ?

guix time-machine

Qu'est-ce que se passe ici ?

```
guix time-machine --no-channel-files \  
  --commit=ee92b0fb7dfd8b55ea63254a59a1605fb870a44c -- \  
  shell --container gcc-toolchain mumps bash coreutils -- bash
```

guix time-machine

Qu'est-ce que se passe ici ?

```
guix time-machine --no-channel-files \  
  --commit=ee92b0fb7dfd8b55ea63254a59a1605fb870a44c -- \  
  shell --container gcc-toolchain mumps bash coreutils -- bash
```

Utilisation d'autres canaux ?

Pokolenie 1 22. máj 2024 02:30:17 (súčasné)

guix-hpc dd9eadb

zdroj repozitára: <https://gitlab.inria.fr/guix-hpc/guix-hpc.git>

vetva: master

úprava: dd9eadb6bd5b73912812594251cb7135327f84c0

...

guix 413ef75

zdroj repozitára: <https://git.savannah.gnu.org/git/guix.git>

vetva: master

úprava: 413ef75f89ac337f7ac3331a862c7c8cdc88aa64

guix time-machine --channels

guix describe

```
Pokolenie 1      22. máj 2024 02:30:17      (súčasn )
guix-hpc dd9eadb
  zdroj repozit ra: https://gitlab.inria.fr/guix-hpc/guix-hpc.git
  vetva: master
   prava: dd9eadb6bd5b73912812594251cb7135327f84c0
guix 413ef75
  zdroj repozit ra: https://git.savannah.gnu.org/git/guix.git
  vetva: master
   prava: 413ef75f89ac337f7ac3331a862c7c8cdc88aa64
```

guix time-machine --channels

guix describe

```
Pokolenie 1      22. máj 2024 02:30:17      (súčasn )
  guix-hpc dd9eadb
    zdroj repozit ra: https://gitlab.inria.fr/guix-hpc/guix-hpc.git
    vetva: master
     prava: dd9eadb6bd5b73912812594251cb7135327f84c0
  guix 413ef75
    zdroj repozit ra: https://git.savannah.gnu.org/git/guix.git
    vetva: master
     prava: 413ef75f89ac337f7ac3331a862c7c8cdc88aa64
```

guix describe -f channels > channels.scm

```
(list
  (channel
    (name 'guix-hpc)
    (url "https://gitlab.inria.fr/guix-hpc/guix-hpc.git")
    (branch "master")
    (commit "dd9eadb6bd5b73912812594251cb7135327f84c0"))
  (channel
    (name 'guix)
    (url "https://git.savannah.gnu.org/git/guix.git")
    (branch "master")
    (commit "413ef75f89ac337f7ac3331a862c7c8cdc88aa64")))
```


guix time-machine --channels

```
guix describe -f channels > channels.scm
```

```
(list  
  (channel  
    (name 'guix-hpc)  
    (url "https://gitlab.inria.fr/guix-hpc/guix-hpc.git")  
    (branch "master")  
    (commit "dd9eadb6bd5b73912812594251cb7135327f84c0"))  
  (channel  
    (name 'guix)  
    (url "https://git.savannah.gnu.org/git/guix.git")  
    (branch "master")  
    (commit "413ef75f89ac337f7ac3331a862c7c8cdc88aa64"))))
```

Tout ça pour dire...

```
guix describe -f channels > channels.scm  
guix time-machine --channels=channels.scm -- \  
  shell --container hello -- hello
```

Qu'en est-il de guix shell ?

Ça va encore

```
guix shell --container gcc-toolchain mumps bash coreutils -- bash
```

Par contre

```
guix shell --pure --with-input=pastix-5=pastix-5-mkl \  
  --with-input=mumps-scotch-openmpi=mumps-mkl-scotch-openmpi \  
  --with-input=openblas=mkl --with-input=slurm=slurm@19 \  
  --with-git-url=gcvb=$HOME/src/gcvb \  
  --with-commit=gcvb=40d88ba241db4c71ac3e1fe8024fba4d906f45b1 \  
  --preserve=~SLURM bash coreutils inetutils findutils grep sed \  
  bc openssl python python-psutil gcvb slurm@19 openmpi scab
```

Passage par un fichier Scheme [5]

Manifests [10]

```
guix shell --container gcc-toolchain mumps bash coreutils  
(specifications->manifest  
  (list "gcc-toolchain" "mumps" "bash" "coreutils"))
```

Passage par un fichier Scheme [5]

Manifests [10]

```
guix shell --container gcc-toolchain mumps bash coreutils  
(specifications->manifest  
  (list "gcc-toolchain" "mumps" "bash" "coreutils"))
```

Tout ça pour dire...

```
guix shell --container gcc-toolchain mumps bash coreutils \  
  --export-manifest > manifest.scm  
guix shell --container --manifest=manifest.scm -- bash
```

Vers un environnement logiciel reproductible

```
guix time-machine --channels=channels.scm -- \
  shell --container --manifest=manifest.scm -- bash
```



Ingrédients nécessaires

- 1 descriptif des paquets logiciels à inclure (+ options)
- 2 descriptif des canaux

Démo

- 1 Café Guix, guix time-machine et guix describe
 - <https://gitlab.inria.fr/mfelsoci/cafe-guix>
- 2 OptiTrust
 - <https://github.com/charguer/optitrust>

Pour la route

Retrouvez cette session

- format tutoriel
- peut être fait en autonomie



<https://tinyurl.com/cafe-guix>

Un tutoriel sur GNU Guix

- présenté au ComPAS'2023
- peut être fait en autonomie



<https://tinyurl.com/tuto-guix>

Bibliographie

- [1] P.R. Amestoy et al. “A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling”. In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 15–41.
- [2] P.R. Amestoy et al. “Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures”. In: *ACM Transactions on Mathematical Software* 45 (1 2019), 2:1–2:26.
- [3] *Canaux (Manuel de référence de GNU Guix)*.
https://guix.gnu.org/sk/manual/fr/html_node/Canaux.html.
- [4] Ludovic Courtès and Ricardo Wurmus. “Reproducible and User-Controlled Software Environments in HPC with Guix”. In: *Euro-Par 2015: Parallel Processing Workshops*. Lecture Notes in Computer Science. Aug. 2015, pp. 579–591. URL: <https://hal.inria.fr/hal-01161771/en>.
- [5] R.K. Dybvig and J.P. Hébert. *The Scheme Programming Language, Fourth Edition*. 2009. URL: <https://www.scheme.com/tspl4/>.
- [6] *Invoquer guix describe (Manuel de référence de GNU Guix)*.
https://guix.gnu.org/en/manual/devel/fr/html_node/Invoquer-guix-describe.html.
- [7] *Invoquer guix pull (Manuel de référence de GNU Guix)*.
https://guix.gnu.org/sk/manual/fr/html_node/Invoquer-guix-pull.html.
- [8] *Invoquer guix shell (Manuel de référence de GNU Guix)*.
https://guix.gnu.org/sk/manual/fr/html_node/Invoquer-guix-shell.html.
- [9] *Invoquer guix time-machine (Manuel de référence de GNU Guix)*. https://guix.gnu.org/en/manual/devel/fr/html_node/Invoquer-guix-time_002dmachine.html.
- [10] *Utilisation de base avec des manifestes (Livre de recettes de GNU Guix)*.
https://guix.gnu.org/sk/cookbook/fr/html_node/Utilisation-de-base-avec-des-manifestes.html.
- [11] Nicolas Vallet, David Michonneau, and Simon Tournier. “Toward practical transparent verifiable and long-term reproducible research using Guix”. In: *Nature Scientific Data* 9 (Oct. 2022). DOI: 10.1038/s41597-022-01720-9.