



Guix

Vers une étude expérimentale reproductible avec GNU Guix

Rencontre sur les logiciels libres de recherche

5 octobre 2023

Marek Felšöci

`marek.felsoci@inria.fr`

Mais c'est qui celui-là ?

Parcours universitaire

- Licence et Master en informatique à l'Unistra
- Doctorat en informatique à l'Université de Bordeaux

Stage de fin d'études (ICube)

- implémentation d'une nouvelle structure de contrôle pour le langage C

Thèse (Inria Bordeaux)

- solveurs de grands systèmes linéaires couplés creux/denses à des fins de simulations aéroacoustiques

Post-doc (Inria Nancy / ICube, poste actuel)

- parallélisation automatique de programmes écrits en C et C++

Contexte scientifique



- simulation de la propagation d'ondes acoustiques enveloppant un avion en vol
- résolution de grands systèmes linéaires couplés creux/denses



Situation réelle [1].

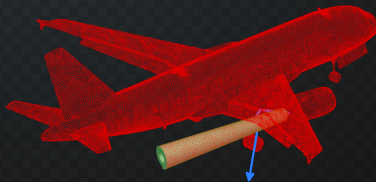
Contexte scientifique

AIRBUS



REGION
Nouvelle-
Aquitaine

- simulation de la propagation d'ondes acoustiques enveloppant un avion en vol
- résolution de grands systèmes linéaires couplés creux/denses



Modèle numérique discret 3D/2D.

Contexte scientifique

AIRBUS


 RÉGION
Nouvelle-
Aquitaine

- simulation de la propagation d'ondes acoustiques enveloppant un avion en vol
- résolution de grands systèmes linéaires couplés creux/denses

$$\begin{bmatrix}
 A_{VV11} & \cdots & A_{VV1n} & A_{SV11}^T & \cdots & A_{SV1k}^T \\
 \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 A_{VVn1} & \cdots & A_{VVnn} & A_{SVn1}^T & \cdots & A_{SVnk}^T \\
 A_{SV11} & \cdots & A_{SV1n} & A_{SS11} & \cdots & A_{SS1k} \\
 \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 A_{SVk1} & \cdots & A_{SVkn} & A_{SSk1} & \cdots & A_{SSkk}
 \end{bmatrix}
 \times
 \begin{bmatrix}
 x_{V_1} \\
 \vdots \\
 x_{V_n} \\
 x_{S_1} \\
 \vdots \\
 x_{S_k}
 \end{bmatrix}
 =
 \begin{bmatrix}
 b_{V_1} \\
 \vdots \\
 b_{V_n} \\
 b_{S_1} \\
 \vdots \\
 b_{S_k}
 \end{bmatrix}$$

Contexte méthodologique

À savoir

- reprendre et enrichir un travail existant
- validation par expérimentation numérique
 - beaucoup d'expériences à prévoir (beaucoup de métriques)
 - sur plusieurs plateformes de calcul haute-performance (HPC)

Contexte méthodologique

À savoir

- reprendre et enrichir un travail existant
- validation par expérimentation numérique
 - beaucoup d'expériences à prévoir (beaucoup de métriques)
 - sur plusieurs plateformes de calcul haute-performance (HPC)

Pour commencer

- construire un cadre d'expériences
- commencer par évaluer l'implémentation actuelle
- valider en reproduisant une partie des expériences précédentes

Liste de courses

- 1 sélectionner un sous-ensemble d'expériences à reproduire
 - choisir algorithmes et métriques à tester

Liste de courses

- 1 sélectionner un sous-ensemble d'expériences à reproduire
 - choisir algorithmes et métriques à tester
- 2 recréer l'environnement expérimental d'origine
 - disponibilité des machines
 - liste des logiciels et des dépendances éventuelles
 - versions utilisées
 - options de construction
 - ...

Liste de courses

- 1 sélectionner un sous-ensemble d'expériences à reproduire
 - choisir algorithmes et métriques à tester
- 2 recréer l'environnement expérimental d'origine
 - disponibilité des machines
 - liste des logiciels et des dépendances éventuelles
 - versions utilisées
 - options de construction
 - ...
- 3 refaire les expériences
 - commandes à lancer
 - paramètres et options d'exécution

Liste de courses

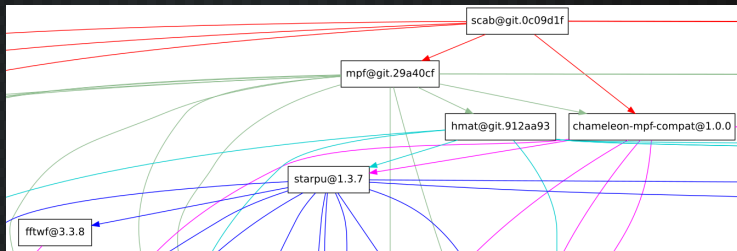
- 1 sélectionner un sous-ensemble d'expériences à reproduire
 - choisir algorithmes et métriques à tester
- 2 recréer l'environnement expérimental d'origine
 - disponibilité des machines
 - liste des logiciels et des dépendances éventuelles
 - versions utilisées
 - options de construction
 - ...
- 3 refaire les expériences
 - commandes à lancer
 - paramètres et options d'exécution
- 4 post-traiter les résultats
 - scripts d'analyse de données
 - génération de figures

Et la reproductibilité dans tout ça ?

Plusieurs problématiques

- 1 reproductibilité de l'environnement matériel
- 2 reproductibilité de l'environnement logiciel
- 3 reproductibilité des expériences elles-mêmes
- 4 conservation à long terme de tous les ingrédients

Défis



- construction : pile logicielle complexe
 - beaucoup de dépendances et de variantes, libres et parfois non-libres
- déploiement : plusieurs machines cibles
 - ordinateur portable, plateformes de HPC
 - PlAFRIM, TGCC, IDRIS, ...
 - disponibilité de logiciels variable

Outils à disposition

Construction manuelle

- télécharger les codes sources nécessaires et compiler manuellement
- trop long et fastidieux

Outils à disposition

Construction manuelle

- télécharger les codes sources nécessaires et compiler manuellement
- trop long et fastidieux

Gestionnaire de paquets ou modules

- s'appuyer sur le gestionnaire de paquets système (ordi. perso.) ou les modules (plateformes HPC)
 - `sudo apt install machin` OU `module load machin`
- disponibilité varie d'une plateforme à l'autre
- pas très modulaire

Outils à disposition

Spack [3]

- gestionnaire de paquets alternatif
- facile de définir de nouveaux paquets et leur variantes
- ne permet pas de garantir la réproductibilité de l'ensemble de l'environnement (compilateurs, ...)



Outils à disposition

Spack [3]

- gestionnaire de paquets alternatif
- facile de définir de nouveaux paquets et leur variantes
- ne permet pas de garantir la réproductibilité de l'ensemble de l'environnement (compilateurs, ...)



Conteneurs (Docker [6], Singularity [5], ...)

- création d'environnements autonomes et reproductibles (sous conditions)
- difficilement modulables sans perdre en réproductibilité
- potentiellement très volumineux



Outils à disposition




Guix

GNU Guix (guix.gnu.org) [2, 8]

- gestionnaire de paquets transactionnel
- descriptions autonomes et exécutables d'environnements logiciels entiers
 - facilement modifiables et reproductibles
 - deux fichiers texte suffisent !

Construction d'un environnement logiciel reproductible avec GNU Guix



python
python-numpy
mon-solveur

- remplacer **OpenBLAS**
par **MKL** partout
- pour « mon solveur »
utiliser le commit
6de5b1c...

Ingrédients nécessaires

- 1 liste de paquets logiciels à inclure (+ options de transformation)
- 2 liste de sources de paquets versionnées

Construction d'un environnement logiciel reproductible avec GNU Guix

python
python-numpy
mon-solveur

- remplacer **OpenBLAS**
par **MKL** partout
- pour « mon solveur »
utiliser le commit
6de5b1c...

+

dépôt officiel « guix »
- commit **1ac4959...**
dépôt « guix-hpc »
- commit **9cc4593...**

Ingrédients nécessaires

- 1 liste de paquets logiciels à inclure (+ options de transformation)
- 2 liste de sources de paquets versionnées

Résumé

Réproductibilité d'environnements logiciels

- un défi de taille
- plusieurs outils à disposition

GNU Guix

- réproductibilité du début à la fin (à travers différentes machines)
- facilement modulable
- présent sur peu de plateformes pour l'instant (PlaFRIM [7], Grid5000 [4])
- solutions de repli pour les plateformes sans Guix

Mon retour d'expérience après 4 années d'utilisation

- la prise en main demande du temps mais ça vaut la peine !
- les ressources pour accompagner les (nouveaux) utilisateurs sont de plus en plus nombreuses

Pour la route

Un tutoriel sur GNU Guix

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

<https://tutoriel-guix-compas-2023.gitlabpages.inria.fr/tutorial/>



Merci pour votre attention !

- [1] David Barrie. *First Airbus A350-900 XWB for Cathay Pacific - take off*. <https://www.flickr.com/photos/curufinwe-xiane/27249076575>. F-WZFX (MSN29 - to be registered as B-LRA) performing final flight before delivery - Toulouse, France (May 25, 2016).
- [2] 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>.
- [3] Todd Gamblin et al. "The Spack Package Manager: Bringing Order to HPC Software Chaos". In: *Supercomputing 2015 (SC'15)*. LLNL-CONF-669890. Austin, Texas, USA, Nov. 2015. DOI: 10.1145/2807591.2807623. URL: <https://github.com/spack/spack>.
- [4] *Grid5000: a large-scale and flexible testbed for experiment-driven research*. <https://www.grid5000.fr/>.
- [5] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. "Singularity: Scientific containers for mobility of compute". In: *PLOS ONE* 12.5 (May 2017), pp. 1–20. DOI: 10.1371/journal.pone.0177459. URL: <https://doi.org/10.1371/journal.pone.0177459>.
- [6] Dirk Merkel. "Docker: Lightweight Linux Containers for Consistent Development and Deployment". In: *Linux Journal* 2014.239 (Mar. 2014). ISSN: 1075-3583.
- [7] *PlaFRIM: Plateforme fédérative pour la recherche en informatique et mathématiques*. <https://plafrim.fr/>.
- [8] 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.