

# Workshop Gricad Grenoble

## Méthodes modernes de déploiement de logiciel HPC

GRICAD - Pôle Calcul  
Bruno Bzeznik

May 30, 2018





## Introduction

<https://deploymenthpc.sciencesconf.org/>



## Aujourd'hui sur les systèmes HPC

Les applications et les bibliothèques sont généralement installées manuellement dans un répertoire partagé sur les noeuds. On crée ensuite des **modules d'environnement** afin de faciliter le chargement d'une version ou d'une autre.

- ▶ Difficile et long: les systèmes de build des applications HPC sont souvent mal documentés, non standards, mal testés, avec des dépendances compliquées,...
- ▶ Problèmes de reproductibilité (des installation et des résultats)
- ▶ Une upgrade système peut tout casser
- ▶ Chaque administrateur fait ce qu'il peut, de manière isolée
- ▶ Les utilisateurs doivent faire avec ce qui est installé dans le système ou dans le répertoire d'applications

# Des questions...

qui mènent à des réponses négatives



Mais ça compile bien chez moi!



On n'a pas les mêmes systèmes ou les mêmes versions de bibliothèques



# Des questions...

qui mènent à des réponses négatives



Mais il y a un paquet pour Debian/Mac/Ma\_distrib\_a\_moi\_que\_j'aime, pourquoi ne l'installez-vous pas?



Quand on installe un paquet, il faut aussi le faire sur tous les noeuds et ce sera le même paquet pour tout le monde, et je ne parle même pas des dépendances... De plus, il n'y a aucune optimisation par rapport à notre matériel



"module" c'est bien  
mais ça ne suffit plus...





- ▶ But principal: automatiser les taches de compilation et faciliter leur répétition, avec pour effet de bord intéressant de consigner, voire documenter, les étapes
- ▶ Eventuellement fournir des binaires tout prêts (sous la forme de modules d'environnement ou de paquets)
- ▶ Permettre le partage et le développement collaboratif

# Des questions...

qui mènent à des réponses zen



Mais ça compile bien chez moi!



Ok, cherchons ensemble ce qui fait planter la compilation de ce paquet dans notre environnement particulier et contribuons à sa portabilité







- ▶ But principal: virtualiser pour s'affranchir du problème de la dépendance à l'OS de base et par effet de bord, permettre le partage facile d'images d'environnements d'exécution
- ▶ Container = virtualisation légère
- ▶ Virtualiser uniquement le contexte d'exécution, mais pas forcément chercher l'isolation totale, qui amène souvent à perdre des performances (ex: virtualisation des accès réseau)
- ▶ MPI pas forcément simple, ou pas containerisé (ex: mpirun singularity ...)

# Des questions...

qui mènent à des réponses zen



Mais il y a un paquet pour  
Debian/Mac/Ma\_distrib\_à\_moi\_que\_j'aime,  
pourquoi ne l'installez-vous pas?



OK, prenons une image Debian,  
installons-y ce paquet et faisons  
la tourner dans un container



**Bon workshop!**