Un retour d'expérience : l'environnement virtuel conda à l'intérieur d'un container singularity pour le calcul HPC

AUDA Yves, AZEMA Philippe, BIGOT Jérôme, CABANAS Laurent, ESCOBAR Joan, GAZEN Didier, GONDET Etienne, GEGOUT Pascal, GRIPPA Manuela, HAGOLLE Olivier, HIERNAUX Pierre, INGLADA Jordi, KERGOAT Laurent

Yves.Auda@get.omp.eu

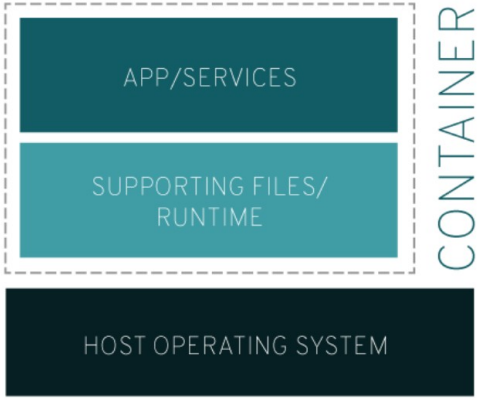
Maching learning / IA

- Télédétection / Landuse
- Chaîne iota2
- Deep learning / keras tensorflow
- Architectures AVX512, GPU
- Outils : gdal, GRASS...
- Langages : R du CRAN, python…
- Multiplicité des versions des outils et langages

Singularity pour le HPC

Un container Linux processus isolé du reste du système (couche LXC)
Possède ses propres fichiers, bibliothèques encapsulés dans une image
Portable sur toutes les machines Linux et uniquement Linux !

La même configuration sur ordinateur personnel et sur un cluster HPC
- bibliothèque optimisées pour architecture du processeur



Mise en œuvre simple

Exemple fichier définition « image.def » d'un container singularity

```
Bootstrap: docker
From: ubuntu:latest
%files
    /home/auda/Téléchargements/Miniconda3-latest-Linux-x86_64.sh /Miniconda3-latest-Linux-x86_64.sh
    /home/auda/.condarc /root/.condarc
%environment
    PATH=/miniconda3/bin:$PATH
%post
    apt-get update && apt-get -y install wget build-essential
    apt-get install locales
    mkdir /workdir
```

Créer/Editer/utiliser le container

Créer le container

```
sudo singularity build –sandbox container.img image.def
```

Modifier le container

```
sudo singularity shell writable container.img
```

Créer une image

```
sudo singularity build image.simg container.img
```

Utiliser l'image avec un point de montage différent du home

```
Singularity -B /pointDeMontage shell image.simg
```

Environnement virtuel conda

- Stabilité système (version python 2,3, pip install...)
- Différentes versions de python
- Différentes versions bibliothèques
- Adaptation des briques de calcul à l’architecture (tensorflow, AVX2, AVX512, GPU)

Environnement virtuel conda

- Stabilité système (version python 2,3, pip install...)
- Différentes versions de python
- Différentes versions bibliothèques
- Adaptation des briques de calcul à l’architecture (tensorflow, AVX2, AVX512, GPU)

Environnement conda

Environnement AVX2 (ordinateur bureau, core i7)

```
conda create --name keras-env
conda install tensorflow
conda install keras -n keras-env
conda install gdal -n keras-env
conda install pandas -n keras-env
conda install matplotlib -n keras-env
```

Pas de paquet spécifique

Environnement tfAVX512 (cluster skylake)

```
conda create --name tfAVX512-env
conda install -c jjh_cio_testing/label/tflow_variants tensorflow-skylake-avx512-base -n tfAVX512-env
...
```

tensorflow-skylake-avx512-base 1.4.1

Environnement GOU (cluster GPU)

```
conda create --name tfGPU-env
conda install tensorflow-gpu==1.12.0 cudatoolkit==9 -n tfGPU-env
...
```

tensorflow-gpu 1.14.0

Conda dans singularity

- Portabilité d’un ordinateur personnel vers un cluster HPC
- Inclure ses programmes
- Inclure chaîne de traitement iota2 (conda)
- Inclure programmes tiers GRASS
- Multiples environnement python 2.7, 3.6, 3.7… gdal2.2.3, 3.0.1...

Test de performance

AVX512 36 coeurs 80 minutes
AVX512 16 coeurs 160 minutes (concurrence autre utilisateur)
n373 CPU 16 coeurs 334 minutes
n373 GPU 16 coeurs 135 minutes