

FORUM HPC XIII 28/11/2023

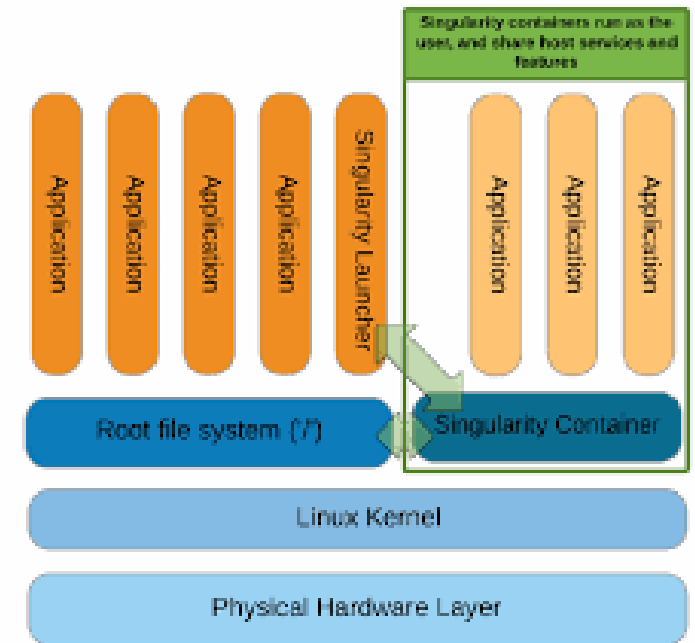
CONTENEURISATION SUR
LE HPC DU CNES AVEC
SINGULARITY

www.thalesgroup.com



Introduction - Qu'est ce que la conteneurisation

C'est une technologie qui permet d'exécuter une application linux à l'intérieur d'un environnement isolé, appelé conteneur, qui ne dépend que du noyau linux de la machine sur laquelle on est.



Introduction - Intérêt de la conteneurisation pour le HPC

- **Diffusion** facile de logiciels, sans procédures d'installation complexes , sans dépendances avec le système hôte.

- **Portabilité** : Compatible simultanément avec différents systèmes d'exploitation hôtes.

Clusters	Système d'exploitation
TREX /ARYX	Linux RedHat 8.x
HAL /KTULU	Linux CentOS 7.x

- **Facilité** de mise en œuvre

Introduction - Choix de SINGULARITY

- **Meilleur compromis , du point de vue sécurité et compatibilité HPC**

suite à un audit fait en 2022 des solutions Docker, Singularity, Apptainer, Sarus et PodMan.



- **Compatible avec les catalogues d'images Docker**

(ex : la forge logicielle du CNES , Artifactory).

- **Solution développée originellement pour le HPC**

Compatible avec les cartes GPUs, les librairies infiniband (Mellanox).

Vocabulaire

➤ **Image** : Description du conteneur, que l'on peut échanger facilement.

Singularity possède son propre mécanisme d'images : statique au format SIF et accessible en écriture (sandbox)

Interface avec des images docker que l'on peut convertir en format SIF.

➤ **Conteneur** : Machine virtuelle légère qui lance une application dans un environnement isolé. Un conteneur, instancié par le lancement d'une image, utilise le noyau Linux de l'hôte qui l'héberge, contrairement à une machine virtuelle.

➤ **OCI** (Open Container Initiative) : Norme sur les conteneurs (format, commandes...)

Singularity - Instanciation d'une image

- Ouvrir un shell en interactif dans le conteneur avec « singularity shell »

```
trexvisu01 # module load singularity/3.10.2

trexvisu01 # singularity shell /softs/rhall/singularity-images/CentOS-7.9_6G.sif

# exit pour sortir du conteneur.
Singularity > exit
```

- Lancer une commande depuis le conteneur avec « singularity exec »

```
trexvisu01 # singularity exec CentOS_7.9_6G.sif cat /etc/redhat-release
CentOS Linux release 7.9.2009 (Core)

# Hors du conteneur ici sur le noeud trexvisu01 , le système est en Linux RedHat8.6
trexvisu01 # cat /etc/redhat-release
Red Hat Enterprise Linux release 8.6 (Ootpa)
```

Singularity - Option --bind

- Utiliser l'option --bind (ou -B) pour monter des espaces disques externes au conteneur.

```
trexvisu01 # singularity shell -B /nfs CentOS_7.9_6G.sif  
  
Singularity > ls /nfs  
data2 gohs1 gohs2 gohs3 gohs4 sgc ssalto
```

- Sans l'option -B /nfs , l'espace /nfs n'est pas visible du conteneur.

```
trexvisu01 # singularity shell CentOS_7.9_6G.sif  
  
Singularity > ls /nfs  
ls: cannot access '/nfs': No such file or directory
```


Singularity - le mode « build » pour construire des images

- Différentes méthodes possibles : depuis un dépôt distant (dockerhub, singularity) , un dépôt CNES (artifactory), une image locale, ou un fichier de définition .def (équivalent dockerfile).

```
# dépôt officiel singularity
singularity build centos8_singularity.sif library://library/default/centos:latest

# dépôt docker
singularity build centos8_docker.sif docker://centos:latest

# image locale
singularity build <container.sif> <directory/image.sif>

# fichier de définition .def
singularity build container.sif definition_file.def
```


Singularity - le mode « fakeroot » de build

- Permet le build d'une image, en se faisant passer pour root.

```
trexvisu01 # module load singularity  
trexvisu01 # export SINGULARITY_TMPDIR=/singularity  
trexvisu01 # singularity build --fakeroot image.sif image.def
```

- **Restriction sur TREX** : Le build se fait uniquement depuis les nœuds trexvisu de TREX

Singularity - le build à partir d'un fichier de définition

– Contient plusieurs sections :

%Header : Méthode utilisée : docker, localimage, yum, hub singularity...

%Files : Pour monter des fichiers externes lors de la construction de l'image.

%Post : installation de bibliothèques/applications dans l'image.

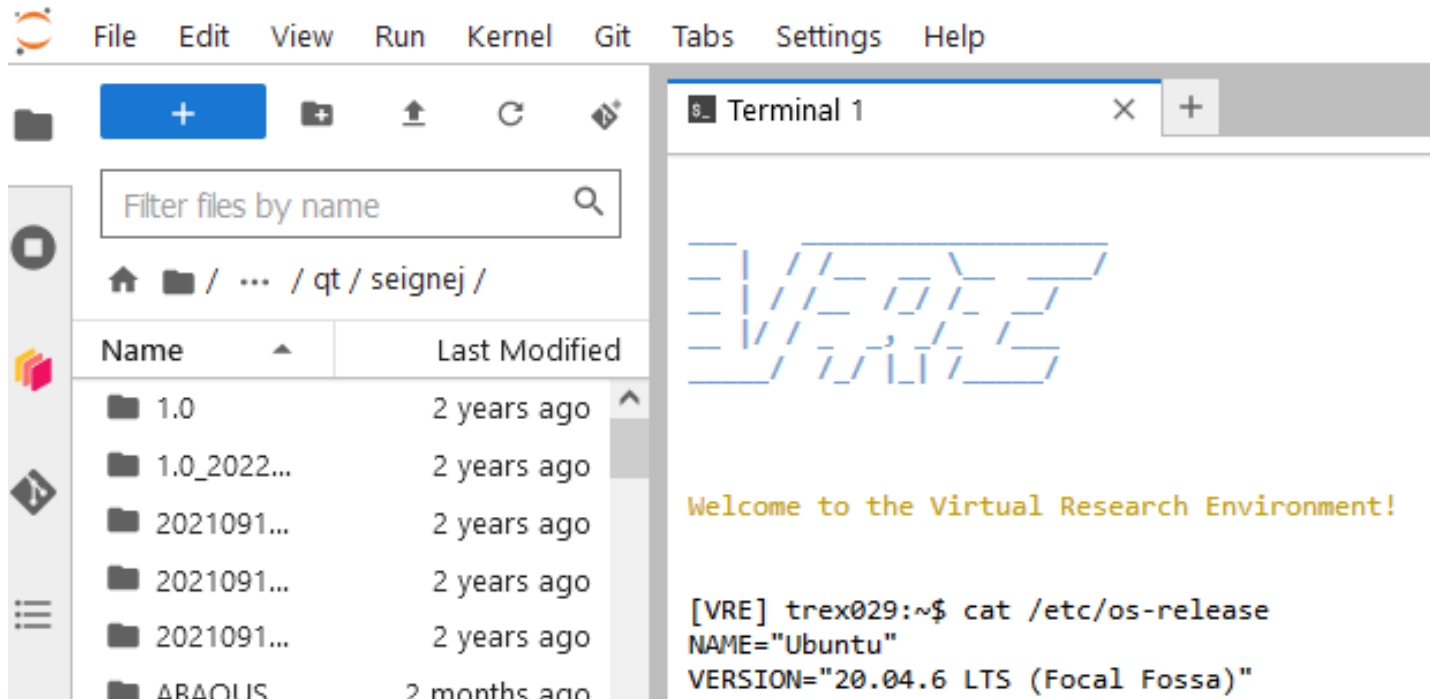
%Environment : Définition de l'environnement à l'exécution de l'image

– **Astuce** : « singularity inspect -d » récupère le fichier de définition d'une image

```
trexvisu01 # singularity inspect -d <image.sif>
```

Singularity – Quelques exemples d'utilisation sur le HPC du CNES

- Une session Datalabs dans une VRE (VRECNES, VREOT, VREAI4GEO ou VRESWOT) est un job SLURM qui lance un conteneur « Ubuntu » sur le cluster TRES.



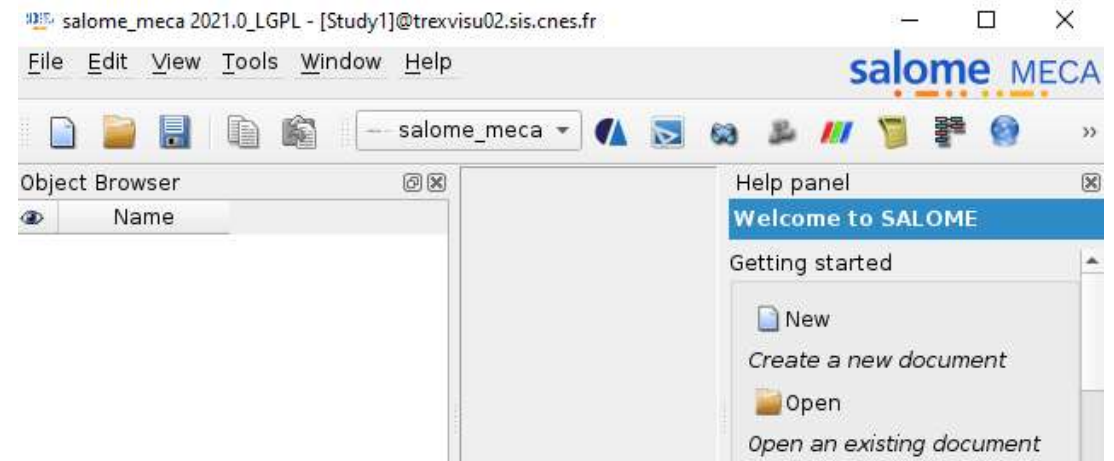
Singularity – Quelques exemples d'utilisation sur le HPC du CNES

- Le module « salome_meca » (plateforme de calcul scientifique libre) est un programme conteneurisé sous Debian9

```
trexvisu02 # module load salome_meca
load singularity/3.10.2 : OK
load salome_meca/2021.0.0-2 : OK

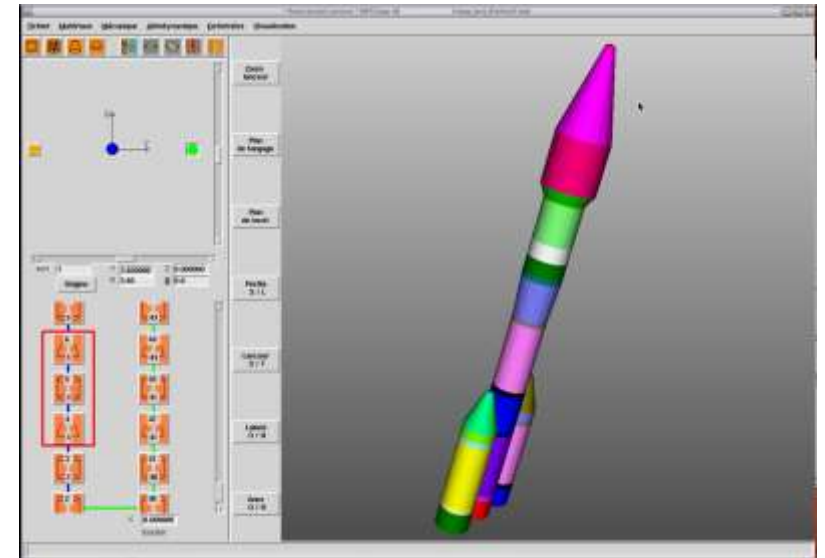
trexvisu02 # salome --shell

Singularity> cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
```



Singularity – Quelques exemples d'applications conteneurisées pour STS

- **Conteneurs systèmes RH7** créés pour l'exécution des applications sous RH8 :
- **SuiteCarmen** : Contient les applications Carins, Cardim et Carfonc
- **Clapp** : Avec intégration des bibliothèques graphiques OpenGL.



Singularity - Utilisation avancée en calcul // avec SLURM

➤ Calcul non conteneurisé

job SLURM // sur 2 nœuds g2019 RH7.

```
#SBATCH --job-name=openmpi_rh7
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH -C 2019

# load default OpenMPI environnement
module load openmpi/4.1.2

# Launch OpenMPI program with srun
srun ./wave_openmpi_rh7.exe
```

Temps de calcul: 130 secondes

➤ Même calcul en **environnement conteneurisé**

Job sur 2 nœuds RH8 g2022 dans un conteneur CentOS7.9

```
#SBATCH --job-name=containerized
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH -C 2022

# load singularity
module load singularity

# Launch OpenMPI program into the container with srun
srun singularity exec CentOS_7.9_6G.sif ./wave_openmpi_rh7.exe
```

Temps de calcul : 80 sec car exécution sur des nœuds g2022

Démo : Build d'une image pour TRES

- Analyse du fichier de définition du conteneur CentOS_7.9_6G.sif , qui installe openmpi-4.1.2 de TRES et les bibliothèques infiniband de TRES:

```
trexvisu03 # singularity inspect -d /softs/rhall/singularity-images/CentOS_7.9_6G.sif
```

- Montrer comment à été construite l'image

```
trexvisu03 # module load singularity  
trexvisu03 # export SINGULARITY_TMPDIR=/singularity  
trexvisu03 # singularity build --fakeroot CentOS_7.9_6G.sif CentOS_7.9_6G.def
```


Démo : Build d'une image pour TRES

- Ouvrir un shell interactif et regarder le contenu de l'image : son système d'exploitation, l'emplacement de l'installation d'openmpi/4.1.2.

```
trexvisu03 # singularity shell /softs/rhall/singularity-images/CentOS_7.9_6G.sif
```

```
Singularity> cat /etc/redhat-release
```

- CentOS Linux release 7.9.2009 (Core)

```
Singularity> ls /opt/openmpi/4.1.2
```

```
bin etc include lib64 share
```

```
Singularity> which mpif90
```

```
/opt/openmpi/4.1.2/bin/mpif90
```

```
Singularity> exit
```



Merci

www.thalesgroup.com