

# TD2 GLCS : HDF5

18/12/2020

## Rendu des TDs

Vous devrez rendre vos TDs par email à l'adresse `julien.bigot@uvsq.fr` avant dimanche soir (20 décembre), minuit. Attention, votre mail devra comporter en fichier attaché une archive (format `tar.gz`) de votre projet ne contenant que les fichiers source, **pas d'exécutable** !

## Mise en place

Commencez par installer docker sur votre machine, puis autorisez votre utilisateur à le lancer. Sous debian / ubuntu, on obtient ce résultat avec les commandes :

```
sudo apt install docker.io
sudo gpasswd -a $USER docker
```

Une fois docker installé, téléchargez le fichier du TD : [https://raw.githubusercontent.com/jbigot/glcs\\_2020-2021/master/glcs-td2.sh](https://raw.githubusercontent.com/jbigot/glcs_2020-2021/master/glcs-td2.sh) puis exécutez le. Cette exécution peut être un peu lente ; elle télécharge tout l'environnement du TD. Elle devrait créer un répertoire TD2 avec les fichiers du TD et ouvrir un shell dans ce répertoire.

```
wget https://raw.githubusercontent.com/jbigot/glcs_2020-2021/master/glcs-td2.sh
bash glcs-td2.sh
```

Le shell ainsi ouvert tourne dans un environnement docker (container) qui comporte des bibliothèques, outils, etc... différents de ceux de votre machine personnelle. C'est dans cet environnement que le TD va se dérouler. Pour vous assurer que votre shell courant tourne bien dans l'environnement du TD, vérifiez que le prompt commence bien par `glcs@`.

Vous pouvez à tout moment ouvrir un nouveau shell dans cet environnement en exécutant la commande :

```
bash glcs-td2.sh <chemin>/<vers>/TD2
```

## 1 Écriture parallèles

Cet exercice reprend l'exercice 5 du TD1.

Dans `ex1/ex1.cxx`, on cherche à rendre le code indépendant de la parallélisation. On veut donc que le fichier généré ne dépende plus du nombre de rang MPI mais seulement de la taille totale `block_width × MPI_size`. Pour cela, il faudra utiliser HDF5 parallèle et créer une fenêtre ("hyperslab" HDF5) non seulement pour sélectionner les données à écrire en mémoire mais aussi pour sélectionner où les écrire dans le fichier.

Modifiez le code pour que tous les rangs MPI (processus) écrivent en parallèle dans un même fichier HDF5 leurs blocs de données dans un tableau dont la largeur totale est `block_width × MPI_size`.

Vous pourrez valider votre code en comparant les fichiers générés en variant le nombre de processus. Les données devraient être les mêmes, quel que soit le nombre de processus MPI. Pensez à tester votre code avec différents paramètres de taille et différents parallélismes MPI.