

XIOS (Travaux pratiques)

0) Installation de XIOS sur la machine – Extraire et compiler

1) XIOS – Ecriture des données

- a. Fichier de configuration iodef.xml – include file
- b. Mode multiple/one file
- c. Création d'une nouvelle grille
- d. Utilisation d'une nouvelle grille – Ajout d'un champ
- e. Opérations temporelles
- f. Opérations arithmétiques
- g. Chaînage d'opérations temporelles
- h. Time series
- i. Lecture des données

2) XIOS – Compression

- a. Compression HDF5
- b. Grilles compressées

3) XIOS - Performances

- a. Performances
- b. Memoire

4) XIOS - Grilles - Transformations

- a. Zoom
- b. Générations de grilles - Interpolations

La session de Travaux Pratiques a pour objectif de vous faire découvrir quelques-unes des fonctionnalités de XIOS 2 sur des cas simples et concrets. Les parties 1, 2, 3 et 4 sont indépendantes : les sources et fichiers xml nécessaires à chacune des parties sont dans des répertoires distincts.

Un conseil pour ces TPs : utilisez la documentation en ligne pour les utilisateurs (non complète) :



http://forge.ipsl.jussieu.fr/ioserver/attachment/wiki/WikiStart/XIOS_user_guide.pdf

0) Installation de XIOS - Extraire et compiler XIOS

Se connecter sur votre machine de calcul.

Extraire XIOS dans un répertoire de travail. XIOS est hébergé sur le serveur svn forge de l'IPSL.

```
mkdir WORK_XIOS ; cd WORK_XIOS ;  
svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk XIOS ;  
cd XIOS ;
```

Compiler XIOS (--help pour avoir les options disponibles, --avail pour connaître les architectures supportées,).

```
# Connaître les options/modes de compilation disponibles  
./make_xios --help (or ./make_xios -h) ;  
# Connaître les architectures/machines supportées  
./make_xios --avail ;  
# A quoi ressemble les fichiers arch ?  
cat arch/arch_X64_CURIE.*  
# Compiler (par exemple Curie)  
./make_xios --prod --arch X64_CURIE --job 4  
# Verifier que la librairie libxios.a et l'executable xios_server.exe ont bien ete cree  
ls bin/xios_server.exe lib/libxios.a
```

1) XIOS – Installation des TPs

Récupérer et installer le fichier http://dods.extra.cea.fr/work/p86caub/XIOS_TAR.tar sur votre machine de calcul. Détarer le fichier et chacun des sous-fichiers tarés. Vous obtenez au final 4 sous-répertoires, chacun correspondant aux 4 parties du TP : **TP_intro**, **TP_comp**, **TP_perfs**, **TP_grilles**. Chaque sous-répertoire est indépendant et contient un Makefile très simple que vous devez adapter en fonction de votre machine de calcul en indiquant en particulier le nom du compilateur et les variables XIOS_DIR, NETCDF_INC_DIR, NETCDF_LIB_DIR (si elles ne sont pas déjà renseignées dans votre environnement).

2) XIOS – Ecriture des données

Se placer dans le répertoire TP_intro. Le programme TP_intro.f90 est l'application de test de la librairie XIOS que vous allez utiliser dans le cadre de ces Travaux Pratiques. C'est l'équivalent de votre « cher et tendre » modèle que vous utilisez habituellement. Editez le programme et observez les différentes parties. Puis compiler ce programme.

```
cd TP_intro  
# Editer le programme TP_intro.f90 (emacs/vi/gedit)  
vi TP_intro.f90  
gmake
```

a. Include file

Le fichier iodef.xml est le fichier de configuration principal de XIOS.

```
# Editer le fichier xml (emacs/vi/nedit)
vi iodef.xml
```

Dans le fichier iodef.xml, activer le contexte « atmosphere » en décommentant la ligne correspondante.



La fonctionnalité « src » est très pratique pour splitter les fichiers xml en plusieurs fichiers selon les définitions du contexte, du domaine, des champs, des fichiers,...

Dans cet exemple, « src="./context_atmosphere.xml" » signifie que les spécifications relatives au contexte atmosphere seront dans le fichier « context_atmosphere.xml ». Editez le fichier « context_atmosphere.xml » et observez sa composition :

- les différentes parties liées au « field », « file », « domain » et « grid ».
- les attributs des « field » et « file »

```
# Exécuter le programme de test (par défaut le programme est lancé sur 8 cœurs de calculs en mode
attaché pour XIOS).
lsubmit Job
```

b. Mode multiple/one file

Observez ce que vous obtenez : nombre de fichiers, nombre de records dans le fichiers,...

```
ls output_atmosphere_2D* ; ncdump -h output_atmosphere_2D_0.nc
```

Combien avez-vous de fichiers « output_atmosphere_2D* » ? Passez en mode « one_file » en modifiant le fichier « context_atmosphere.xml » et relancer le Job. Combien de fichiers obtenez-vous ? Vérifiez que le champ est correct : vous pouvez visualiser un des champs obtenus (par exemple la variable « tsol ») à l'aide de ferret :

```
ferret
yes?use output_atmosphere_2D.nc
yes?shade TSOL[l=1]
```



Attention la fonctionnalité “one_file” ne peut fonctionner que si la librairie netcdf/HDF5 utilisée a été compilée pour tourner en mode « parallèle ».

c. Création d'une nouvelle grille

Les champs « tsol », « flat » et « slp » sortis dans le fichier « output_atmosphere_2D.nc » sont des champs 2D utilisant la grille « grid_atm_2D » définie dans le fichier « context_atmosphere.xml » par

le domaine 2D « domain_atm ». On veut maintenant sortir des champs 3D sur une grille composée du domaine 2D « domain_atm » (déjà existant) et d'un axe vertical « axis_atm ». Il faut donc :

- créer ce nouvel axe vertical « axis_atm » dans le fichier « context_atmosphere.xml » dans une section « axis_definition ».
- ajouter (en décommentant) la définition de cet axe dans le modèle (ici le programme test_TP.f90).
- créer une nouvelle grille composée du domaine « domain_atm » et de l'axe vertical « axis_atm » dans le fichier « context_atmosphere.xml ». Cette nouvelle grille s'appellera : «grid_atm_3D »

d. Utilisation d'une nouvelle grille

On veut sortir les nouveaux champs « temp » et « rhum » sur la nouvelle grille. Il faut donc :

- définir les champs dans le fichier xml. Les caractéristiques de ces champs sont les suivants :
 - id="temp" name="temp" long_name="Air temperature" unit="K" grid_ref="grid_atm_3D"
 - id="rhum" name="rhum" long_name="Relative humidity" grid_ref="grid_atm_3D"
- activer l'envoi de ces champs dans le modèle test_TP.f90
- activer la sortie de ces champs dans un fichier avec les caractéristiques du fichier suivantes :
 - id="output_atmosphere_3D" name="output_atmosphere_3D"

```
# Compiler
gmake
# Exécuter le programme de test
lsubmit Job
```

Vérifier avec « ferret » ou « ncdump » que les nouveaux champs ont bien été sortis dans le fichier « output_atmosphere_3D ».

e. Opérations temporelles

Les champs sortis par défaut sont des valeurs moyennées tous les jours. Modifier le fichier xml (et en particulier les attributs « operation » et « output_freq ») afin de sortir les valeurs des champs :

- moyennées toutes les heures
- instantanées à tous les pas de temps
- maximales de chaque journée

Vérifier le résultat en observant le nombre de record dans les fichiers obtenus, par exemple :

```
ncdump -h output_atmosphere_2D.nc
```

f. Opérations scalaires

Utiliser les opérations scalaires afin de sortir les champs de température en degrés Celsius (au lieu de degrés Kelvin par défaut). Il s'agit de sortir dans le fichier « output_atmosphere_3D.nc » un nouveau champ qui hérite des attributs du champ « temp » avec les caractéristiques suivantes :

- `name="tempC" long_name="Air temperature en degC" unit="degC"`

Pour vous aider, regarder dans le User's Guide la partie : « How to use arithmetic opérations ». Vérifier les résultats en plottant (ferret) la variable obtenue « tempC ».

g. Chaînage des opérations temporelles

Utiliser les opérations temporelles afin de sortir dans le fichier « *output_atmosphere_3D.nc* » la moyenne sur 5 jours du maximum journalier du champ « temp ». Ce nouveau champ héritera des attributs du champ « temp » et aura les caractéristiques suivantes :

- `field_ref="temp" name="temp_ave_max"`

Utiliser pour cela le champ intermédiaire

- `field id="temp_max" field_ref="temp" operation="maximum" grid_ref="grid_atm_3D"`

Pour vous aider, regarder dans le User's Guide la partie : « How to chain multiple temporal operations ». Vérifier les résultats en plottant (ferret) la variable obtenue « temp_ave_max ».

h. Times series

Sortir les champs « flat », « tsol » et « slp » sous forme de Times Series (une variable par fichier). On veut que ces champs soient sortis :

- à la fois en Time Series ET dans leur fichier originaux « *output_atmosphere_2D.nc* »
- dans des fichiers commençant par « TS »
- dans des fichiers splittés tous les jours.

Utiliser pour cela les attributs de fichier suivants :

- `timeseries="both"`
- `ts_prefix="TS"`
- `ts_split_freq="1d"`

et l'attribut de champ

- `ts_enabled=".true."`

Les options possibles pour l'attribut de fichier "timeseries" sont

- none : pas de fichier TS.
- both : fichier TS pour les champs qui ont l'attribut « `ts_enabled=".true."` » et sorties de tous les champs (`ts_enabled = .true.` et `ts_enabled = .false.`) dans le fichier brut.
- only : seulement les fichiers TS pour les champs qui ont l'attribut « `ts_enabled=".true."` ».
- exclusive : fichiers TS pour les champs qui ont l'attribut « `ts_enabled=".true."` » et champs qui n'ont pas l'attribut « `ts_enabled=".true."` » sortis dans le fichier brut.



Un fichier « *xios_registry.bin* » permet de gérer les timeseries lors d'un arrêt/reprise d'un run. Ce fichier est à manipuler comme un fichier de restart (output en fin de run, input en début du run)

suivant). Il est créé au cours du run et permet à XIOS de connaître le nom des fichiers « timeseries » à compléter au cours de la simulation lors d'une reprise de run.



Afin d'avoir des « times series » qui s'ajoutent dans le même fichier au cours de la simulation, il faut activer le mode « append » du fichier : `append= « true »`.

i. Lecture des données

La fonctionnalité de lecture des champs dans un fichier est une nouvelle fonctionnalité de XIOS2. Rajouter la lecture de la variable « tsol » dans la boucle temporelle depuis le fichier TS « TS_sol.nc » précédemment écrit (copier le fichier « TS_sol.nc » en « TS_sol_saved.nc » afin de pouvoir l'utiliser). La fonction à appeler dans le modèle est la fonction « xios_recv_field ». Ecrivez la variable lue en TS dans un nouveau fichier « TS_sol_read.nc ». A l'aide la commande `ncdiff` comparer le fichier obtenu avec le fichier initial `TS_sol_saved.nc`.

```
TP_intro.f90 : CALL xios_recv_field("tsol_recv",src_field_recv_tsol(,:))
<file id="tsol_recv" name="TS_sol_saved" output_freq="1ts" mode="read" enabled=".true.">
  <field id="tsol_recv" freq_offset="1ts" name="tsol" grid_ref="grid_atm_2D" operation="instant" enabled=".true." />
</file>
```



Par défaut, la 1^{ère} lecture se fait au pas de temps 0 : pour faire commencer la lecture au pas de temps 1 (1^{er} indice de la boucle temporelle) utiliser l'attribut du champ «`freq_offset="1ts" »`.

3) Compression

```
cd ~/TP_comp
# Editer le programme TP_comp.f90 (emacs/vi/gedit)
vi TP_comp.f90
gmake
```

a. Compression HDF5

La compression gzip disponible dans HDF5 permet de réduire la taille des fichiers de sorties. L'attribut de champ « `compression_level` » permet d'activer cette compression dans XIOS. Le niveau de compression disponible est compris entre 0 à 9 (0 = pas de compression). Utiliser cet attribut pour sortir votre fichier « `output_atmosphere_2D.nc` » au mode compressé de niveau 2. Comparer la taille de vos fichiers avec et sans compression. Augmenter la fréquence d'output pour mieux observer l'effet de la compression.



La compression HDF5 n'est pas disponible en parallèle, ce qui signifie que vous devez tourner en mode « `multiple_file` » ou bien en mode serveur mais avec un seul serveur.

b. Grilles compressées

La fonctionnalité « Compression with Gathering (Section 8.2. Compression by Gathering, CF convention) permet de stocker dans le fichier NetCDF les seules valeurs non masquées en éliminant les points manquants (missing values) des variables. Cette fonctionnalité est disponible dans XIOS2 et permet entre autres :

- de réduire la taille du fichier en éliminant les points masqués.
- de pouvoir sortir plusieurs points du domaine (stations,...) dans un seul et même fichier.

Utiliser le tableau « src_mask » (dans le programme TP_comp.f90) pour définir un masque du domaine « domain_atm » grâce à la fonction « xios_set_domain_attr » et l'argument « mask_1D ». Sortir la variable tsol en utilisant cette compression dans un nouveau fichier « output_atmosphere_2D_compressed.nc ». Pour cela, il suffit d'utiliser l'attribut de champ « indexed_output=".true." » pour le champ « tsol » dans le fichier xml.

```
gmake ; Isubmit Job
```

Observez le fichier final et repérer l'attribut NetCDF « compress = "lat lon" ».

4) Performances - Mémoire

Intéressons-nous aux performances d'écritures de la librairie XIOS.

Pour cette partie sur les performances, les runs tourneront sur le \$SCRATCHDIR (cf dans le Job : RUNDIR=\$SCRATCHDIR/RUN_{\$NCPUS_CLIENT}CLIENT_{\$NCPUS_SERVER}SERVER dans le Job).

```
cd ~/TP_perfs ; gmake
```

- 1) Par défaut dans ce programme de test, le mode est « **attaché** » (pas de serveur) et « **multiple_file** » : les écritures se font en mode synchrone par les 8 clients qui écrivent chacun dans un seul fichier en séquentiel.

```
Isubmit Job
```

Observer les rapports sur les temps fournis par XIOS dans les fichiers xios_client_0.out ainsi que le temps de restitution du run.

- 2) Passez en mode « **one_file** » (toujours en mode **attaché**) comme attribut pour l'ensemble des fichiers: les écritures se font en mode synchrone par les 8 clients qui écrivent chacun dans un seul et même fichier en parallèle.

```
Isubmit Job
```

Observer les rapports sur les temps fournis par XIOS dans les fichiers xios_client_0.out ainsi que le temps de restitution du run. Quelles sont les différences par rapport au mode « multiple_file » ?

- 3) Restez en mode « **one_file** » ET passez en mode **serveur**. Pour cela, il suffit de lancer l'exécutable du modèle ET l'exécutable du serveur (mode MPMD) : décommenter les lignes pour cela dans le Job, ajuster le nombre de cœurs pour le serveur (variable NCPUS_SERVER) ainsi que le nb de cœurs demandés dans l'entête du Job. Les écritures se font ainsi en mode asynchrone par le serveur.

lsubmit Job

Observer les rapports sur les temps et les ratios fournis par XIOS dans les fichiers xios_client_0.out et xios_server_0.out ainsi que le temps de restitution du run. Quelles sont les différences obtenues par rapport au mode attaché ?

- 4) Restez en mode « **one_file** » ET passez en mode **serveur** sur **2, 4, 6, 8,...serveurs**. Pour cela, ajustez la variable NCPUS_SERVER dans le Job et le nb de cœurs demandés dans l'entête du Job. Les écritures se font ainsi en mode asynchrone par les serveurs dans un seul et même fichier en parallèle.

lsubmit Job

Observer les rapports sur les temps et les ratios fournis par XIOS dans les fichiers xios_client_0.out et xios_server_0.out ainsi que le temps de restitution du run. Quelle est la configuration optimale obtenue ?

- 5) Par défaut le mode « **performance** » est activé. Activer le mode « **memory** » dans le fichier iodef.xml, relancer le job et analyser les différences en mode « serveur ». Qu'observez vous entre les mémoires des buffers mode « memory » et mode « performance » ? Qu'observez vous entre les temps obtenus en mode « memory » et mode « performance » ?

« memory » : optimisation mémoire : les buffers peuvent traiter un seul champ à la fois.

« performance » : optimisation performance = gain temps de restitution : les buffers peuvent traiter l'ensemble des champs d'un pas de temps d'écriture



5) XIOS - Grilles - Transformations

Intéressons-nous aux transformations de grilles.

```
cd ~/TP_transf ; gmake
```

a. Zooms

Sortir la variable « tsol » sur une partie de la grille « grid_atm_2D » qui sera un carré de 10 points sur 10 points démarrnant aux premiers indices i et j. Cette variable sera sortie dans un nouveau fichier

« output_atmosphere_2D_zoom.nc ». Vous créez pour cela une nouvelle grille « grid_atm_2D_zoom » composée d' un nouveau domaine qui s'appellera « domain_atm_zoom » : ce domaine fera référence au domaine global « domain_atm » et utilisera les attributs relatifs au zoom.

```
<grid id="grid_atm_2D_zoom">
  <domain domain_ref="domain_atm" >
    <zoom_domain id="domain_atm_zoom" ibegin="0" jbegin="0" ni="10" nj="10" />
  </domain>
</grid>
```



Les indices des tableaux utilisés par XIOS démarrent à 0.



La fonctionnalité de zoom peut être utilisée pour sortir dans un fichier une valeur sur un seul point de la grille.

lsubmit Job

Observez la variable tsol dans le fichier « output_atmosphere_2D_zoom.nc ».

b. Génération de grilles - Interpolations

L'utilisation d'interpolations est une fonctionnalité nouvelle dans XIOS2. Sans ne rien changer au code, on veut pouvoir :

- sortir la variable tsol dans un fichier « output_atmosphere_2D.nc » sur sa grille d'origine
- sortir la variable tsol sur une grille régulière de résolution 360 x 180 dans un fichier « output_atmosphere_2D_HR.nc ». Pour cela il faut créer une nouvelle grille grid_atm_2D_HR composée du domaine « domain_atm_HR » qui sera le résultat de l'interpolation du « domain_atm » (domaine origine) sur le « domain_atm_HR » (nouveau domaine HR). Cette interpolation se définit en 2 étapes/filtres :
 - o génération de la nouvelle grille régulière HR
 - o interpolation sur la nouvelle grille HR

Ces 2 filtres définissent se définissent de la façon suivante :

```
<domain id="domain_atm_HR" ni_glo="360" nj_glo="180" type="rectilinear">
  <generate_rectilinear_domain />
  <interpolate_domain/>
</domain>

<grid id="grid_atm_2D_HR">
  <domain domain_ref="domain_atm_HR" />
</grid>
```

Il suffit ensuite de définir un nouveau fichier comportant le nouveau champ sur la nouvelle grille :

```
<field field_ref="tsol" grid_ref="grid_atm_2D_HR" />
```

lsubmit Job

Visualiser la variable « t_{sol} » obtenues sur les 2 grilles différentes. Refaites la même chose en sortant la variable t_{sol} sur une grille à une résolution différente. Visualiser le résultat.