

Rappel des bases de GNU/Linux pour les TDs numériques

Anne-Sophie Mouronval - Filippo Gatti

Laboratoire MSSMat UMR CNRS 8579, CentraleSupélec

22 février 2022

Follow this tutorial :

- 1 Connexion à un server de calcul
- 2 La ligne de commande
- 3 Moyens de calcul du Mésocentre Moulon
 - Installer et utiliser python
- 4 Gestions de jobs au Mésocentre Moulon
- 5 Visualisation des résultats scientifiques

Connexion à un server de calcul

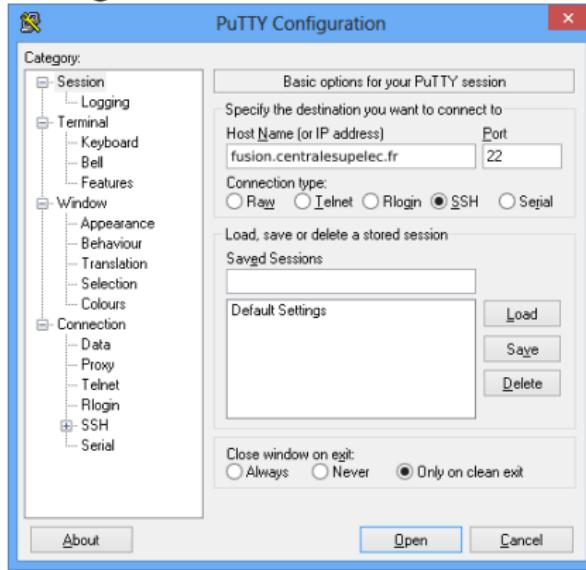
Sur windows, on peut installer le terminal ubuntu :

<https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6?activetab=pivot:overviewtab>



Connexion à Ruche avec PuTTY

- Téléchargement/installation de PuTTY (<https://putty.org>)
- Configuration de la connexion :



- Dans l'onglet *Menu*
 - set *Enable TCP keepalives*
 - set *30s between keepalives*

- La commande ssh permet de se connecter à un compte sur une machine distante sous GNU/Linux (ou Mac OS X) à partir d'une machine sous GNU/Linux (ou Mac OS X).
- Exemple : se connecter au compte comptedst sur la machine distante machinedst.centralesupelec.fr (hostname affiche le nom de la machine où l'on se trouve)

```
mnd@machine ~$ ssh comptedst@machinedst.centralesupelec.fr  
comptedst@machinedst.centralesupelec.fr password:  
  
comptedst@machinedst ~$ hostname  
machinedst
```

- L'option -X de ssh permet d'avoir le retour graphique (utile lorsqu'on utilise un logiciel ou un éditeur avec une interface graphique sur la machine distante).
- Lorsque l'on souhaite se connecter à une machine distante sous GNU/Linux depuis une machine sous Windows, on utilise des outils comme puTTY (<http://www.putty.org/>).

Connexion à Ruche sous GNU/Linux

- 3 comptes sont disponibles par groupe :
invsem0X - passwords : ask the teacher !
 - X : de 1 à 9
 - ① Groupe 1 : invsem01 à invsem03
 - ② Groupe 2 : invsem04 à invsem06
 - ③ Groupe 3 : invsem07 à invsem09

```
mnd@machine~$ ssh -X invsem0X@ruche.mesocentre.universite-paris-saclay.fr  
  
invsem0X@ruche.mesocentre.universite-paris-saclay.fr password:  
*.....*  
: user: invsem0X - group: teaching - project: invsem  
: accounting on last 30 days: Jobs: 0 - Walltime(h) x Ncpus: 0  
: home     quota: 0 / 10G (0%) files: 10 / 100,000 (0%)  
: workdir  quota: 0 / 200G (0%) files: 1 / 200,000 (0%)  
*.....*  
invsem0X@ruche01~$ hostname  
ruche
```

- Doc Ruche :

https://mesocentre.pages.centralesupelec.fr/user_doc

Read the doc!

The screenshot shows a two-column layout. The left column is a sidebar with a dark background containing a navigation menu. The right column is the main content area.

Mesocentre Documentation

Docs » Home

Moulon mesocentre

The foundations of this project rest upon the decision of two institutions, CentraleSupélec and ENS Paris-Saclay, to pool their computing resources within the Paris-Saclay University, host them at IDRIS and set up a common support team.

This project was motivated by two reasons :

- The first one concern economies of scale, extensibility of the system and eco-responsible pooling.
- The second reason is to allow the experimentation, the adjustment and then the consolidation of the establishment of shared computing resources between institutions, with a particular attention given to (i) the organization of an efficient and proximity support structure, gathering engineers and technicians currently in charge of the HPC facilities on the two establishments and (ii) the setting up of a joint scientific animation based on the exchange of skills and know-how between the different disciplines.

The mesocentre is intended to increase its computing capabilities and open itself to researchers from other institutions as well as industrial companies.

It is funded by the Contrat Plan État Région (CPER).

Figure –

https://mesocentre.pages.centralesupelec.fr/user_doc/

Transfert de fichiers

- La commande `scp` permet de transférer des fichiers ou des répertoires sur une machine distante sous GNU/Linux (ou Mac OS X) à partir d'une machine sous GNU/Linux (ou Mac OS X).

```
$ scp [-r] origin destination
```

- Exemple : transférer un fichier `ficloc` se trouvant dans `/home/mnd/tp` sur la machine locale vers la machine distante `machinedst.centralesupelec.fr` dans le répertoire `/workdir/comptedst/rep` du compte `comptedst`. Notez que le mot de passe du compte distant est demandé avant le transfert.

```
mnd@machine ~$ #la commande est écrite sur 2 lignes mais doit
mnd@machine ~$ #être tapée sur une seule ligne
mnd@machine ~$ scp /home/mnd/tp/ficloc
comptedst@machinedst.centralesupelec.fr:/workdir/comptedst/rep/
comptedst@machinedst.centralesupelec.fr password:
ficloc 100% 555KB 555.1KB/s 00:00
```

Transfert de fichiers

- Exemple : transférer le fichier `ficdst` se trouvant sur la machine distante dans le répertoire `/workdir/comptedst/rep` vers la machine locale dans `/home/mnd/tp`

```
mnd@machine ~$ #la commande est écrite sur 3 lignes mais doit
mnd@machine ~$ #être tapée sur une seule ligne
mnd@machine ~$ scp
    comptedst@machedst.centralesupelec.fr:/workdir/comptedst/rep/ficdst
    /home/mnd/tp/
comptedst@machedst.centralesupelec.fr password:
ficdst 100% 555KB 555.1KB/s 00:00
```

- L'option `-r` de `scp` permet de transférer des répertoires.
- Lorsque l'on souhaite transférer des fichiers vers une machine distante sous GNU/Linux depuis une machine sous Windows, on utilise des outils comme WinSCP (<https://winscp.net>) ou PSCP (téléchargeable depuis <http://www.putty.org/>).

La ligne de commande

Le shell et le terminal

- Le shell est le programme qui interprète les commandes et les transmet au noyau Linux.
- Le terminal permet d'interagir avec le shell.
- A l'ouverture du terminal, le shell s'initialise en exécutant des scripts et affiche une invite de commande (*prompt*).

```
gattif@ruche01.centralesupelec.fr's password:  
Last login: Thu Feb  4 16:15:57 2021 from 138.195.213.28  
  
Bonjour !  
the Mésocenter team wishes you  
a great time computing on ruche.  
We appreciate your feedback.  
-----  
Support : ruche_support@groupes.renater.fr  
Website : http://mesocentre.centralesupelec.fr/  
https://mesocentre.pages.centralesupelec.fr/user\_doc/  
[gattif@ruche01 ~]$ █
```

Syntaxe générale

```
$ command_name [options] [arguments]
```

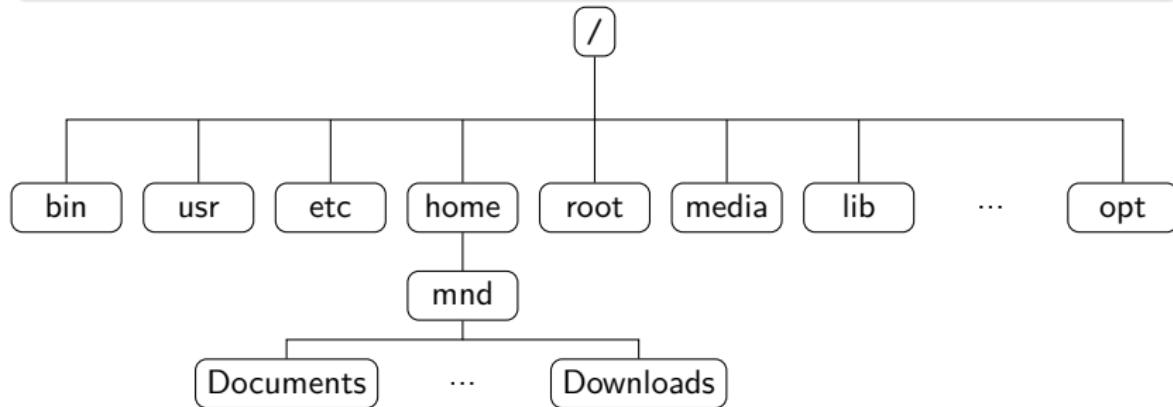
- Le caractère séparateur entre les différents composants d'une commande est le blanc (<SPACE>).
- Les options (lettres-clés facultatives) modifient le comportement de la commande et sont en général précédées du signe moins "-".
- Les arguments (parfois facultatifs) spécifient les objets sur lesquels la commande s'applique.
- Attention : le shell est sensible à la casse (MAJ/min) !

Exemples

```
mnd@machine ~$ rm -rf Rep1 fichier2  
mnd@machine ~$ cp -r Rep1 NewRep1
```

Le système de fichiers et la structure arborescente

- Un système de fichiers (*File System*) est une structure de données permettant de stocker les informations et de les organiser dans des fichiers sur des mémoires secondaires (disques durs...).
- Le système de fichiers est organisé en une structure arborescente dont la racine est représentée par /.



Exemple d'une arborescence de machine sous GNU/Linux

Les commandes cd et pwd

- La commande `pwd` affiche le nom du répertoire courant (où l'on se trouve dans l'arborescence) :

```
mnd@machine ~$ pwd  
/home/mnd
```

- La commande `cd` permet de se déplacer dans l'arborescence :

```
mnd@machine ~$ pwd  
/home/mnd  
mnd@machine ~$ cd Documents  
mnd@machine ~/Documents$ pwd  
/home/mnd/Documents
```

- La commande `cd` sans argument ou `cd ~` permet de se déplacer vers son *homedir* :

```
mnd@machine ~/Documents$ cd  
mnd@machine ~$ pwd  
/home/mnd
```

Les commandes cd et pwd (suite)

- Le déplacement se fait en utilisant le chemin absolu ou relatif :

```
mnd@machine ~$ pwd  
/home/mnd  
mnd@machine ~$ cd Downloads  
mnd@machine ~/Downloads$ pwd  
/home/mnd/Downloads  
mnd@machine ~/Downloads$ cd  
mnd@machine ~$ pwd  
/home/mnd  
mnd@machine ~$ cd /home/mnd/Downloads  
mnd@machine ~/Downloads$ pwd  
/home/mnd/Downloads
```

- La commande cd .. permet d'aller au répertoire parent :

```
mnd@machine ~/Downloads$ pwd  
/home/mnd/Downloads  
mnd@machine ~/Downloads$ cd ..  
mnd@machine ~$ pwd  
/home/mnd
```

Syntaxe générale des noms de fichiers et de répertoires

- Un nom de fichier ou de répertoire peut contenir, en général, jusqu'à 255 caractères.
- Il y a sensibilité à la casse (min/MAJ).

Conseils

- Eviter d'utiliser les espaces, les caractères spéciaux comme *, ?, &, !, ‘, /, \$, [,] et les lettres accentuées.
- Se limiter aux caractères alphabétiques minuscules ou majuscules (de a à z et de A à Z), aux chiffres (de 0 à 9), au point (.), au caractère de soulignement "_" et au tiret "-".
- Exemples :
 - `un_nom_de_fichier.txt`
 - `UnNomDeRepertoire`

Les éditeurs de texte

Un éditeur de texte est un logiciel destiné à la création et l'édition de "fichiers de texte purs" (à ne pas confondre avec "traitement de texte").

Les 2 plus utilisés

- **vi, sera utilisé pour les TDs sur Ruche**
 - Editeur standard Unix (très présent)
 - Utilisé le plus souvent en mode ligne de commande
 - Variantes : vim (VI Improved) et gvim (mode graphique)
- Emacs (Editing MACroS)
 - Editeur GNU
 - Utilisable en ligne de commande
- Ces 2 éditeurs sont efficaces mais complexes au premier abord.

Autres éditeurs (plus simples mais moins répandus)

- Nedit (sous GPL, multi-plateformes)
- Gedit (éditeur Gnome)
- Mousepad (éditeur XFCE)

Utilisation de base de vi

- `vi file` ouvre le fichier `file` (existant ou non)
- `vi` dispose de 2 modes : le mode commande et le mode insertion
- A l'ouverture du fichier, le mode commande est actif
- Pour saisir du texte, passez en mode insertion en tapant `i` (`INSERT` s'affichera en bas du terminal) et taper le texte. Utilisez `[ESC]` pour sortir de ce mode et revenir au mode commande.
- En mode commande ou insertion, les flèches du clavier permettent de se déplacer dans le fichier (ceci peut se faire aussi en mode commande à partir de commandes spécifiques à `vi`)
- Pour effacer le caractère sous le curseur, en mode commande, tapez `x` puis `i` pour revenir en mode insertion
- Pour effacer une ligne complète, en mode commande, tapez `dd`
- Pour annuler les effets de la commande précédente, en mode commande, tapez `u`
- Pour sauvegarder le fichier, en mode commande, tapez `:w`
- Pour sauvergarder le fichier et sortir de `vi`, en mode commande, tapez `:x`
- Pour sortir de `vi` sans sauvergarder le fichier, en mode commande, tapez `:q!`
- Pour plus de commandes, consultez <https://fr.wikipedia.org/wiki/Vi>

La commande ls

- La commande ls permet de lister le contenu d'un répertoire :

```
mnd@machine ~$ ls
Desktop  Documents  Downloads  file1  Music  Pictures
Public   Templates  Videos
```

- L'option -l permet d'obtenir une liste détaillée (type, droits, propriétaire, taille, date...) :

```
mnd@machine ~$ ls -l
total 32
drwxr-xr-x 2 mnd mnd 4096 sept. 15 18:57 Desktop
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Documents
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Downloads
-rw-rw-r-- 1 mnd mnd    32 sept. 17 17:16 file1
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Music
drwxr-xr-x 2 mnd mnd 4096 sept. 17 10:59 Pictures
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Public
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Templates
drwxr-xr-x 2 mnd mnd 4096 sept. 14 23:08 Videos
```

Les commandes mkdir et rm

- La commande `mkdir` permet de créer un répertoire :

```
mnd@machine ~$ mkdir Tp1
mnd@machine ~$ ls
Desktop  Documents  Downloads  file1  Music  Pictures
Public   Templates  Tp1        Videos
```

- La commande `rm` permet d'effacer des fichiers ou des répertoires :

```
mnd@machine ~$ rm file1
mnd@machine ~$ ls
Desktop  Documents  Downloads  Music  Pictures
Public   Templates  Tp1        Videos
```

- L'option `-r` est nécessaire pour effacer un répertoire :

```
mnd@machine ~$ rm -r Tp1
mnd@machine ~$ ls
Desktop  Documents  Downloads  Music  Pictures
Public   Templates  Videos
```

La commande cp

- La commande cp permet de copier des fichiers ou des répertoires :

```
mnd@machine ~$ ls
Desktop Documents Downloads file1 Music Pictures
Public   Templates Tp1          Videos
mnd@machine ~$ cp file1 file2
mnd@machine ~$ ls
Desktop Documents Downloads file1 file2 Music
Pictures Public    Templates Tp1      Videos
```

- L'option -r est nécessaire pour copier un répertoire :

```
mnd@machine ~$ cp -r Tp1 Tp2
mnd@machine ~$ ls
Desktop Documents Downloads file1 file2 Music
Pictures Public    Templates Tp1     Tp2     Videos
```

La commande mv

- La commande `mv` permet de renommer ou de déplacer des fichiers ou des répertoires :

```
mnd@machine ~$ ls
Desktop  Documents  Downloads  file1  Music  Pictures
Public   Templates  Tp1        Videos
mnd@machine ~$ mv file1 file2
mnd@machine ~$ ls
Desktop  Documents  Downloads  file2  Music  Pictures
Public   Templates  Tp1        Videos
mnd@machine ~$ mv file2 Tp1
mnd@machine ~$ ls
Desktop  Documents  Downloads  Music  Pictures
Public   Templates  Tp1        Videos
mnd@machine ~$ ls Tp1
mnd@machine ~$ file2
```

La commande tar

L'archivage permet de rassembler plusieurs fichiers en un seul afin de faciliter leur manipulation.

La commande tar permet de gérer l'archivage.

- créer une archive (option -c) en la nommant (option -f)

```
mnd@machine ~$ ls
file1.txt  file2.txt  Dir
mnd@machine ~$ tar -cf archive.tar file1.txt file2.txt
mnd@machine ~$ ls
archive.tar  file1.txt  file2.txt  Dir
mnd@machine ~$ #Exemple avec un répertoire
mnd@machine ~$ ls Dir
file3.txt file4.txt
mnd@machine ~$ tar -cf archive2.tar Dir
mnd@machine ~$ ls
archive.tar  archive2.tar  file1.txt  file2.txt  Dir
```

La commande tar (suite)

- afficher le contenu d'une archive (option -t)

```
mnd@machine ~$ tar -tf archive.tar
file1.txt
file2.txt
mnd@machine ~$ tar -tf archive2.tar
Dir/
Dir/file3.txt
Dir/file4.txt
```

- défaire une archive (option -x)

```
mnd@machine ~$ ls
archive.tar  archive2.tar
mnd@machine ~$ tar -xf archive.tar
mnd@machine ~$ ls
archive.tar archive2.tar file1.txt file2.txt
mnd@machine ~$ tar -xf archive2.tar
mnd@machine ~$ ls
archive.tar archive2.tar file1.txt file2.txt Dir
```

- penser à afficher le contenu d'une archive avant de la défaire
- utiliser l'option -v (mode verbeux)

Les commandes gzip et gunzip

La compression est utile pour réduire la durée des transferts par le réseau et pour gagner de l'espace disque (quotas...).

- Les commandes gzip et gunzip permettent respectivement de compresser et décompresser un fichier :

```
mnd@machine ~$ ls
file1.txt file2.txt
mnd@machine ~$ gzip file1.txt
mnd@machine ~$ ls
file1.txt.gz file2.txt
mnd@machine ~$ gunzip file1.txt.gz
mnd@machine ~$ ls
file1.txt file2.txt
```

- La commande gzip est souvent utilisée conjointement avec la commande tar via l'option -z :

```
mnd@machine ~$ tar -zcvf archive.tar.gz file1.txt file2.txt
mnd@machine ~$ ls
archive.tar.gz file1.txt file2.txt
```

La commande man

- La commande `man` permet d'obtenir de l'aide sur les commandes.
- Les touches `↑`, `↓` et `<SPACE>` permettent de faire défiler le texte et la touche `q` permet de quitter.

```
mnd@machine ~$ man ls
LS(1) User Commands          LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILEs (the current directory by default).
(...)
-a, --all
      do not ignore entries starting with .
(...)
```

Doc Ruche :

https://mesocentre.pages.centralesupelec.fr/user_doc

- home directory (or homedir) : /home/username où vous pouvez stocker vos fichiers source (code, données légères). Ce volume est limité à 10 Go par utilisateur. Vous pouvez vous référer à ce répertoire avec la variable d'environnement \$HOME.
- the working directory (or workdir) : /workdir/username où vous pouvez stocker vos fichiers binaires et vos données. Ce volume est limité à 200 Go par utilisateur. Vous pouvez vous référer à ce répertoire avec la variable d'environnement \$WORKDIR.
- Vous pouvez rapidement vérifier votre quota de fichiers avec la commande `ruche-quota`. Si vous avez besoin de plus d'informations sur l'utilisation de vos fichiers, essayez la commande `mmlsquota`.

Moyens de calcul du Mésocentre Moulon

- Plusieurs composants logiciels (compilateur, bibliothèques, logiciels, etc.) sont déjà disponibles sur le cluster de Ruche.
- La commande module vous permet de personnaliser votre environnement. Chaque module chargé met à jour vos variables d'environnement (telles que votre variable \$PATH) pour rendre un composant logiciel accessible par votre code.

Prise en main des moyens de calcul du Mésocentre Moulon

Exemple : vérifier la présence du module boost

```
[invsem0X@ruche01 ~]$ module spider boost  
-----  
boost/1.69.0: boost/1.69.0/intel-19.0.3.199-intel-mpi-python  
-----  
  
This module can be loaded directly: module load boost/1.69.0/intel-19.0.3.199-intel-mpi-python  
  
Help:  
Boost provides free peer-reviewed portable C++ source libraries,  
emphasizing libraries that work well with the C++ Standard Library.  
Boost libraries are intended to be widely useful, and usable across a  
broad spectrum of applications. The Boost license encourages both  
commercial and non-commercial use.
```

Exemple : load/unload le module intel/19.0.3/gcc-4.8.5 et intel-mkl/2019.3.199/intel-19.0.3.199

```
[invsem0X@ruche01 ~]$ module load intel/19.0.3/gcc-4.8.5 intel-mkl/2019.3.199/intel-19.0.3.199  
[invsem0X@ruche01 ~]$ module unload intel/19.0.3/gcc-4.8.5 intel-mkl/2019.3.199/intel-19.0.3.199
```

Exemple : liste des modules téléchargés :

```
[invsem0X@ruche01 ~]$ module list  
Currently Loaded Modulefiles:  
1) intel/19.0.3/gcc-4.8.5 2) intel-mkl/2019.3.199/intel-19.0.3.199
```

Installation environnement python sur Ruche (1)

- Nous vous recommandons de charger Anaconda pour utiliser python3 sur Ruche. Plusieurs versions sont disponibles :

```
[invsem0X@ruche01 ~]$ module avail anaconda  
----- /gpfs/softs/modules/modulefiles/languages -----  
anaconda2/2019.10/gcc-9.2.0 anaconda3/2020.02/gcc-9.2.0 anaconda3/2021.05/gcc-9.2.0
```

- La version python fournie par un module Anaconda donné apparaît dans la sortie de l'aide du module :

```
[invsem0X@ruche01 ~]$ module help anaconda3/2021.05/gcc-9.2.0  
----- Module Specific Help for "anaconda3/2021.05/gcc-9.2.0" -----  
FIXME: Put a proper description of your package here.
```

- Profile/environnement Anaconda par default :

```
[invsem0X@ruche01 ~]$ module load anaconda3/2021.05/gcc-9.2.0  
[invsem0X@ruche01 ~]$ conda list  
# packages in environment at /gpfs/opt/languages/anaconda3/2021.05:  
#  
# Name           Version        Build  Channel  
_ipyw_jlab_nb_ext_conf    0.1.0          py37_0  
_libgcc_mutex            0.1            main  
alabaster                0.7.12         py37_0  
(...)
```

Installation environnement python sur Ruche (2)

- Créer un profile/environnement Anaconda personnalisé :

```
[invsem0X@ruche01 ~]$ module purge
[invsem0X@ruche01 ~]$ mkdir $WORKDIR/.conda
[invsem0X@ruche01 ~]$ ln -s $WORKDIR/.conda ~/.conda
[invsem0X@ruche01 ~]$ module load anaconda3/2021.05/gcc-9.2.0
[invsem0X@ruche01 ~]$ conda create -n myenv
Solving environment: done
## Package Plan ##
environment location: /home/username/.conda/envs/myenv
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use:
# > source activate myenv
#
# To deactivate an active environment, use:
# > source deactivate
#
```

Installation environnement python sur Ruche (3)

- Activer le profile/environnement Anaconda personnalisé :

```
[invsem0X@ruche01 ~]$ source activate myenv
```

- Installation du package mpi4py :

```
(myenv) [invsem0X@ruche01 ~]$ conda install mpi4py
Solving environment: done
## Package Plan ##
environment location: /home/username/.conda/envs/myenv
added / updated specs:
- mpi4py

The following packages will be downloaded:
package          |      build
-----|-----
(...)           |
mpich2-1.4.1p1   |          0      2.7 MB
python-3.6.6     |      h6e4f718_2    28.9 MB
(...)
```

Example installation python

Installation de pytorch avec cuda sur Ruche

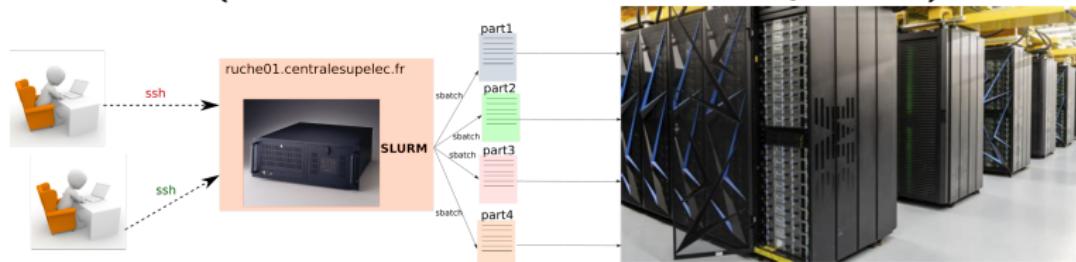
```
[invsem0X@ruche01 ~]$ module load anaconda3/2020.02/gcc-9.2.0
[invsem0X@ruche01 ~]$ module load cuda/9.2.88/intel-19.0.3.199
[invsem0X@ruche01 ~]$ mkdir -p /gpfs/workdir/invsem01/.conda
[invsem0X@ruche01 ~]$ conda create -n myenv -y
[invsem0X@ruche01 ~]$ source activate myenv
[invsem0X@ruche01 ~]$ ln -s /gpfs/workdir/invsem01/.conda ~/.cond
[invsem0X@ruche01 ~]$ conda install pytorch torchvision -c pytorch -y
[invsem0X@ruche01 ~]$ conda install matplotlib -c pytorch -y
[invsem0X@ruche01 ~]$ conda env export > environment.yml # save conda environment
```

On peut créer un fichier `create_conda_env.sh` contenant ces lignes et en suite :

```
[invsem0X@ruche01 ~]$ chmod +x create_conda_env.sh
[invsem0X@ruche01 ~]$ source create_conda_env.sh
```

Gestions de jobs au Mésocentre Moulon

Ne pas exécuter votre code directement sur le terminal de connexion (la *frontale* `ruche01.centralesupelec.fr`) !



- Pour soumettre un travail sur le cluster, il faut décrire les ressources (cœurs, mémoire, temps) dont vous avez besoin au gestionnaire de tâches *SLURM Workload Manager*.
- Le gestionnaire de tâches (SLURM) lancera le travail dès que les ressources sur la **partition** demandée seront disponibles.

SLURM est configuré pour un *partage équitable* :
plus de ressources demandées = priorité inférieure
si le gestionnaire a plusieurs travaux à gérer en même temps.

Partitions sur ruche

CPU partitions

These are the common partitions for CPU computing on ruche.

Resources

Queue name	Nb of nodes	Nb of cores/node	Max. memory per node	Walltime max
cpu_short*	[1-10]	[1-40]	180 GB	01:00:00
cpu_med	[1-10]	[1-40]	180 GB	04:00:00
cpu_long	[1-2]	[1-40]	180 GB	72:00:00
cpu_prod**	[1-50]	[1-40]	180 GB	06:00:00

* : this partition is meant for test and therefore restricted to 1 job per user.

** : the cpu_prod partition is allowed to run jobs :

- at night during weekdays (from 20 pm to 8 am)
- all day long during the weekend

Limits

Queue name	Max Ncpus per user	Max Jobs per user	Max Waltime
cpu_short	600	1	01:00:00
cpu_med	600	-	04:00:00
cpu_long	80	-	72:00:00
cpu_prod	2000	-	06:00:00

Mem partitions

These are the partitions for CPU computing on shared memory on a large number of memory or a large number of cores.

Resources

Queue name	Nb of nodes	Nb of cores/node	Max. memory per node	Walltime max
mem_short*	1	[1-80]	1.5 TB	01:00:00
mem	1	[1-80]	1.5 TB	24:00:00

* : this partition is meant for test and therefore restricted to 1 job per user.

Limits

Queue name	Max Ncpus per user	Max Jobs per user	Max Waltime
mem_short	80	1	01:00:00
mem	80	-	24:00:00

GPU partition

This queue is meant for GPU computing and redirect jobs onto GPU resources.

Resources

Queue name	Nb of nodes	Nb of cores/node	Max. memory per node	Walltime max	GPUs per node
gpu_test*	[1-2]	40	756 GB	01:00:00	4
gpu	[1-2]	40	756 GB	24:00:00	4

* : this partition is meant for test and therefore restricted to 1 job per user.

Limits

Queue name	Max Ngpus per user	Max Jobs per user	Max Waltime
gpu_test	8	1	01:00:00
gpu	8	-	24:00:00

SLURM Command

The `sinfo` SLURM command lists the available partitions for the supercomputer.

Gestion des jobs avec SLURM

Il existe deux façons d'exécuter un code de calcul sur la fusion :

- en utilisant un travail SLURM **interactif** (**pour tests légers et à la configuration de l'environnement**)

CPU :

```
srun --nodes=1 --time=00:30:00 -p cpu_short --pty /bin/bash
```

GPU :

```
srun --nodes=1 --time=06:00:00 -p gpu --gres=gpu:1 --mem=50GB --export=none --pty /bin/bash
```

- en utilisant un **script batch** SLURM à soumettre au planificateur, qui l'exécutera lorsque les ressources seront disponibles (pour les cycles de *production*).

```
sbatch script.sh
```

Pour plus de détails :

https://mesocentre.pages.centralesupelec.fr/user_doc/ruche/06_slurm_jobs_management/

Exemple de script SLURM

- Structure du fichier SLURM

```
#!/bin/bash
#SBATCH -J NomJob
#SBATCH -ntasks=24 ← processus parallèles à allouer
#SBATCH -nodes=1 ← nombre de noeuds
#SBATCH -gres=gpu:2 ← nombre de GPU
#SBATCH -output mo.txt ← fichier de output
#SBATCH -error me.txt ← fichier d'erreur
#SBATCH -time=00:30:00 ← walltime demandé (depende des queues)
#SBATCH -partition gpu ← partition
#SBATCH -mail filippo.gatti@centralesupelec.fr← adresse mail A CHANGER!

#chargement des modules
module purge← nettoyage des modules téléchargés
module load intel-compilers/2018.2← chargement des module nécessaires

cd $SLURM_O_WORKDIR

# lancement du calcul
mpirun -np 48 [-flags mpi] drive.exe [-flags] < input.txt > output.txt
```

Useful commands :

- `squeue -a` : liste des tous les jobs en queue (Q=queued, R=running, H=handle, E=exiting)
- `squeue -u invsem0X` : liste de jobs en queue lancé par `invsem0X`

Visualisation des résultats scientifiques

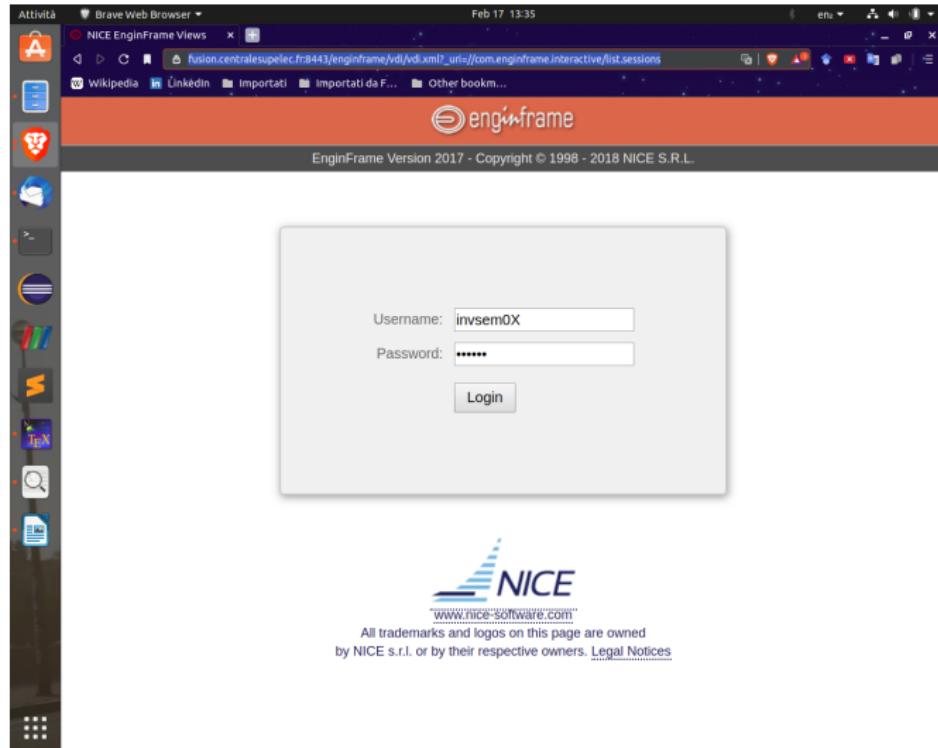
Visualisation des résultats scientifiques

Comment visualiser les résultats (lourds en mémoire !) d'un calcul scientifique ?

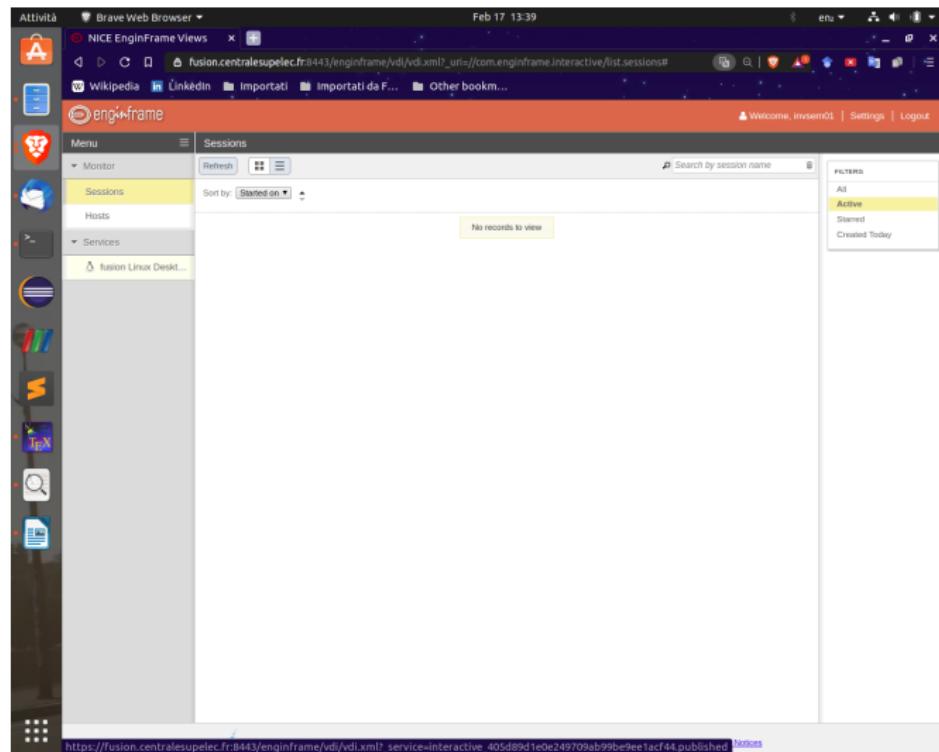
Ruche mets à disposition une **interface de visualisation** sans devoir télécharger les données !

- ① télécharger Nice DCV Client
 - <https://download.nice-dcv.com/>
 - <https://www.realvnc.com/fr/connect/download/viewer/>
- ② Se connecter à https://ruche01.centralesupelec.fr:8443/enginframe/vdi/vdi.xml?_uri=/com.enginframe.interactive/list.sessions
- ③ Ouvrir une session de visualisation (cf diapo d'après)
- ④ Ouvrir le lien avec Nice DCV Client
- ⑤ Lancer Paraview ou autres

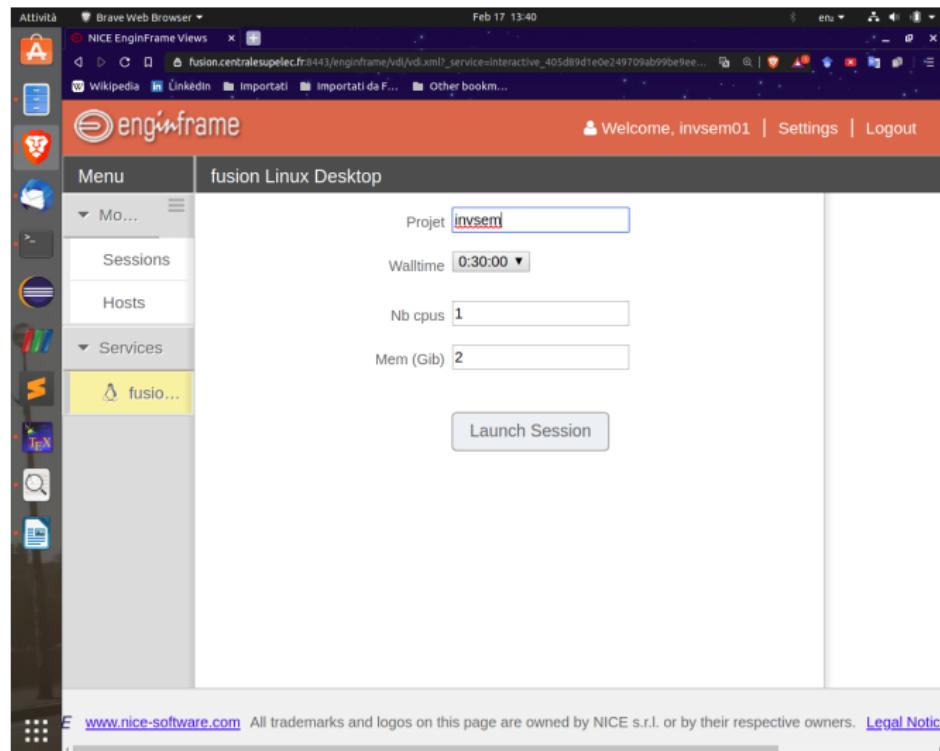
Visualisation des résultats scientifiques



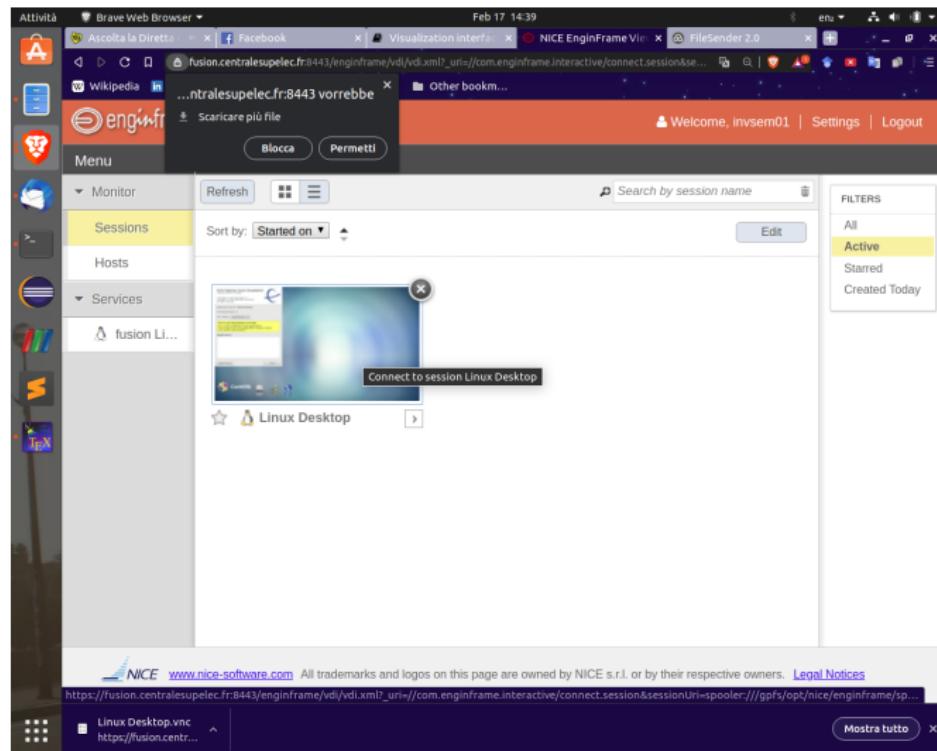
Visualisation des résultats scientifiques



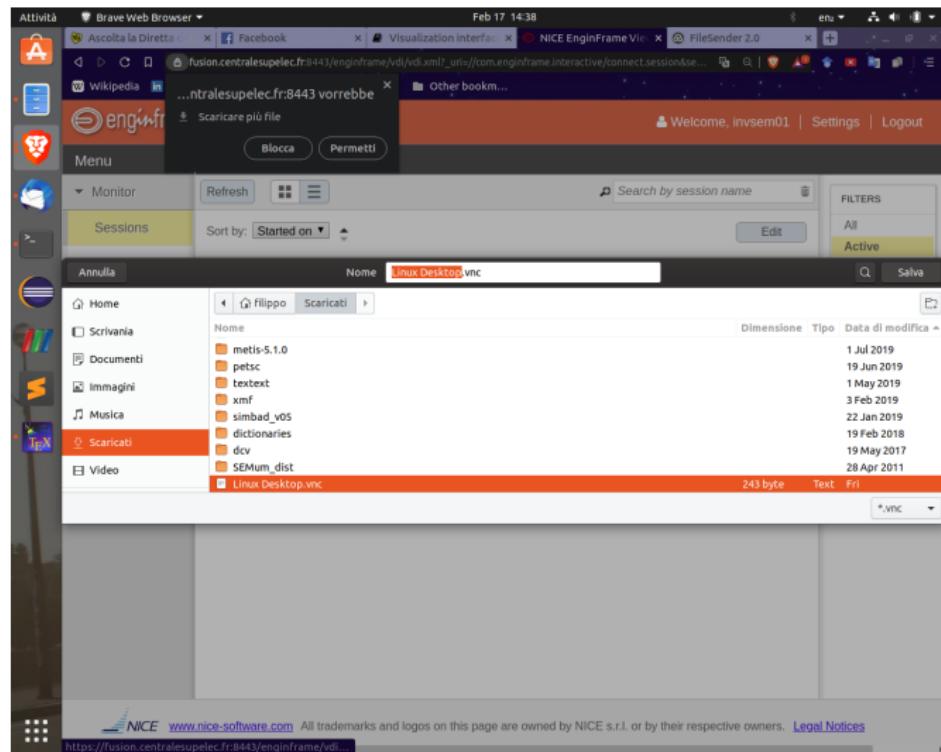
Visualisation des résultats scientifiques



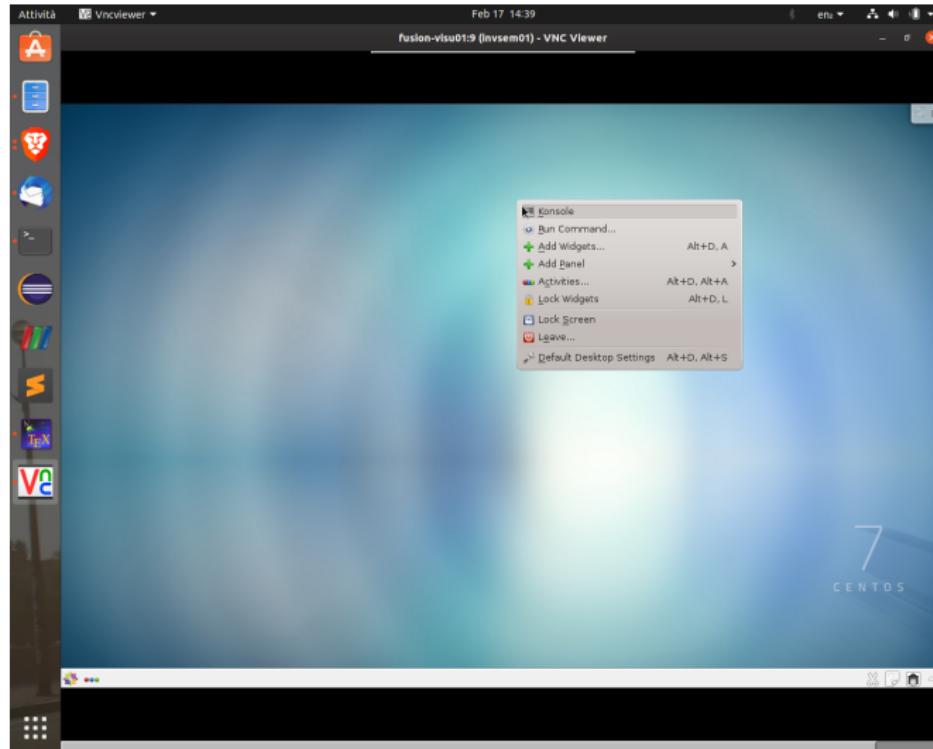
Visualisation des résultats scientifiques



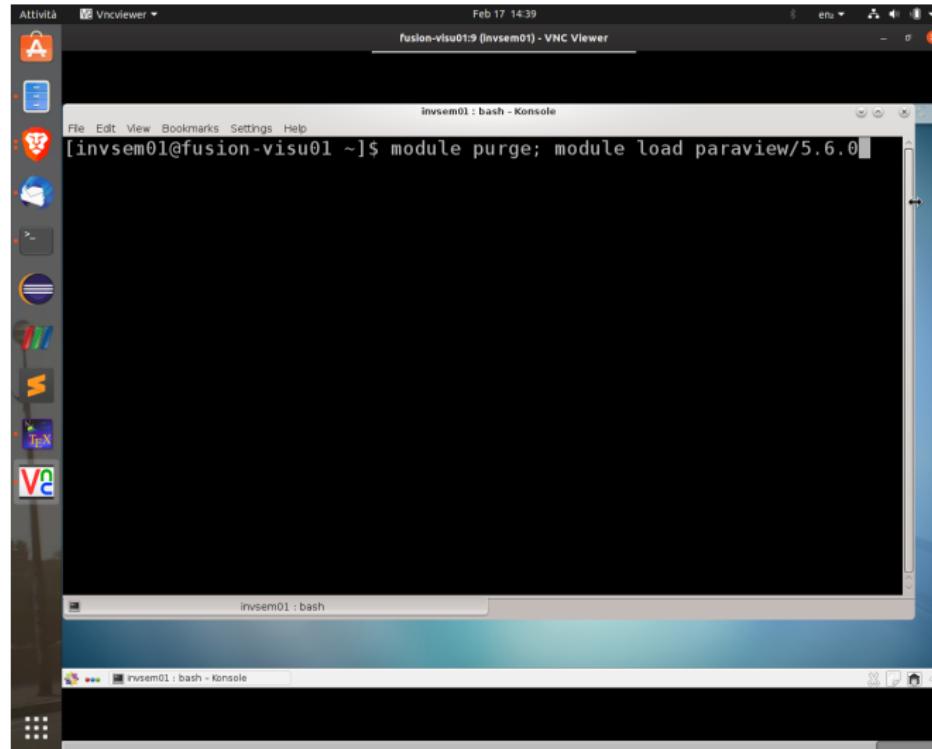
Visualisation des résultats scientifiques



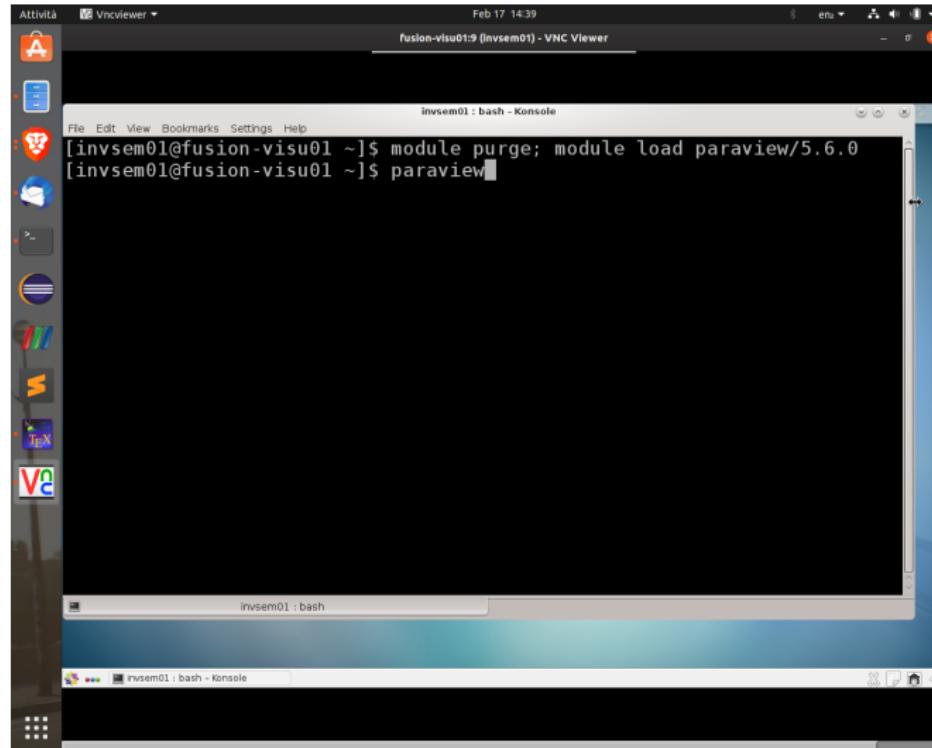
Visualisation des résultats scientifiques



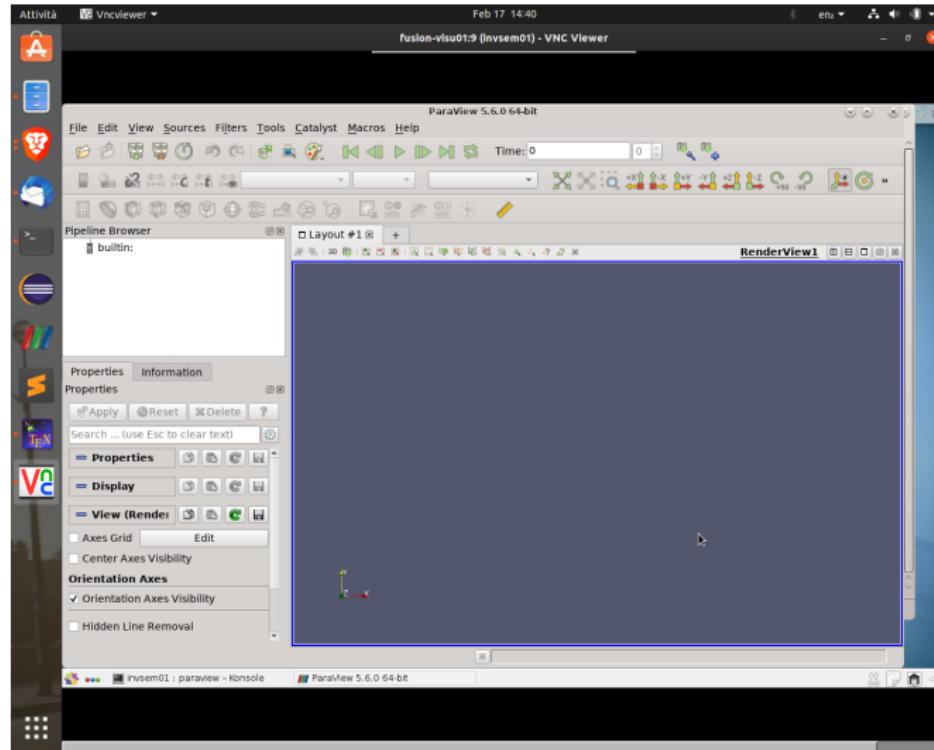
Visualisation des résultats scientifiques



Visualisation des résultats scientifiques



Visualisation des résultats scientifiques



- **Liste de support du mésocentre**

`fusion_support@groupes.renater.fr`

- **Wiki du Mésocentre :**

`https://mesocentre.pages.centralesupelec.fr/user_doc/`

Références

- Cette présentation repose sur "Rappel des bases de GNU/Linux pour les TP", Anne-Sophie Mouronval et Laurent Series, Ecole "Informatique pour le calcul scientifique", 22-26 septembre 2014.
- Formation GNU/Linux disponible sur
<http://mesocentre.centralesupelec.fr/training-fr/unix>
- D'innombrables formations disponibles sur internet