

Introducción_Ometeotl_2025

Ingreso a Clúster y sesión de trabajo.

Elaboró: Pável Ernesto Oropeza Alfaro (poropeza@atmosfera.unam.mx)

Sesión 1

SSH y Bash

- ☐ Ingreso por SSH: `ssh -l usuario_cluster ometeotl.atmosfera.unam.mx -p 9022`
- ☐ Herramientas y comandos UNIX, editores de texto, entorno bash
- ☐ Variables de entorno
- ☐ módulos de software

Breve descripción del Clúster:

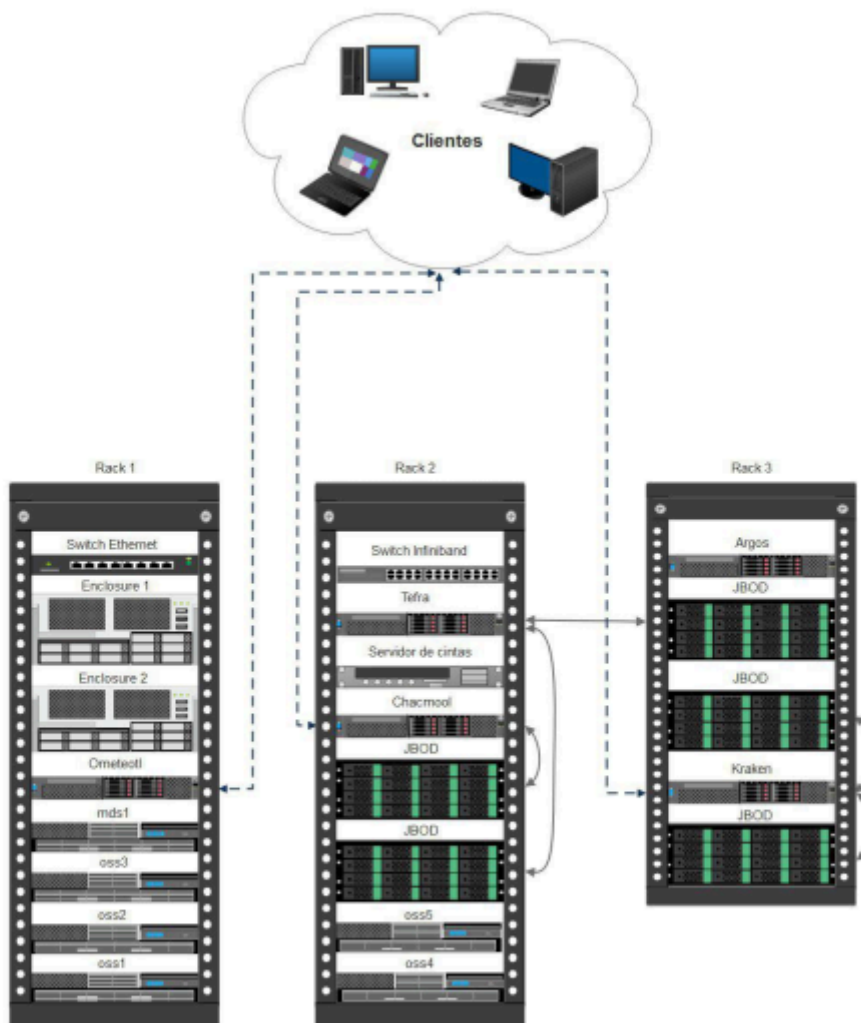


Figure 1: Clúster Omoteotl y servidores de almacenamiento

Software disponible y organización mediante Módulos de software

Para habilitar el entorno de ejecución del modelo WRF es necesario seguir un orden para cargar los módulos que harán que la versión de WRF 4 deseada se encuentre disponible.

A continuación se describe el proceso:

- ☐ Cargar módulos según orden y jerarquía. Se empieza por cargar el compilador:

```
ml load intel/2022u2
```

- ☐ Módulos de bibliotecas requeridas por WRF

```
ml load curl hdf5 jasper libaec libpng mpi netcdf-c netcdf-fortran zlib
```

- ☐ Carga de módulo del modelo WRF, versión 4.2.1

`ml wrf/4.2.1`. Si se desea salvar este orden de carga de módulos de software, ejecutar la siguiente instrucción: `ml save wrf-operativo`. En una sesión posterior, se puede invocar nuevamente a todos los módulos anteriores ejecutando la siguiente instrucción:

```
ml restore wrf-operativo
```

```
{pronostico} ~/WRFV4
(operativo)-[26]: ml list

Currently Loaded Modules:
  1) intel/2022u2/compilers   3) zlib/1.2.12   5) curl/7.82.0   7) jasper/1.900.22   9) netcdf-c/4.8.1   11) wrf/4.2.1
  2) mpi/intel               4) libaec/1.0.6   6) libpng/1.6.37   8) hdf5/1.10.8   10) netcdf-fortran/4.5.4
```

- ☐ Verificación de variables \$PATH y \$LD_LIBRARY_PATH

- ☐ `type -a wrf.exe; echo $PATH`

WRF (ejecutables para experimentos)

- ☐ `geogrid.exe, ungrib.exe, metgrid.exe, real.exe, ndown.exe y wrf.exe`

```
encit00@ometeotl:~$ type -a geogrid.exe ungrib.exe metgrid.exe real.exe ndown.exe wrf.exe
geogrid.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WPS/bin/geogrid.exe
ungrib.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WPS/bin/ungrib.exe
metgrid.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WPS/bin/metgrid.exe
real.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WRFV4/main/real.exe
ndown.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WRFV4/main/ndown.exe
wrf.exe is /LUSTRE/OPERATIVO/OPERATIVO02/modelos2/WRF/WRFV4/main/wrf.exe
```

Sesión 2

A continuación se describe un ejemplo de ejecución del modelo WRF, iniciando desde la etapa del Preprocesamiento (WPS).

- **Estructura de directorios para WRF:**

```
# CREAR LA SIGUIENTE ESTRUCTURA DE DIRECTORIOS Y ARCHIVOS:
cd
mkdir -p ~/WRFV4/{WPS/geogrid,WPS/metgrid}
mkdir ~/WRFV4/WRF
cp /LUSTRE/ID/hidromet/WRF/namelist_pronosticos/4.2.1/namelist.wps
~/WRFV4/WPS
```

```
cp /LUSTRE/ID/hidromet/WRF/namelist_pronosticos/4.2.1/namelist.input
~/WRFV4/WRF

cd ~/WRFV4/WPS/geogrid
ln -sv /LUSTRE/ID/hidromet/WRF/wps_files/geogrid/GEOGRID.TBL .
cd ../metgrid
ln -sv /LUSTRE/ID/hidromet/WRF/wps_files/metgrid/METGRID.TBL .
cd ..
ln -sv /LUSTRE/ID/hidromet/WRF/wps_files/ungrib/Variable_Tables/Vtable.GFS
Vtable
cp /LUSTRE/ID/hidromet/WRF/wps_files/link_grib.csh .
```

Se utiliza la configuración de dos dominios definida en los archivos *namelist.wps* y *namelist.input*:

☐ Ejecutar WPS

Los archivos *namelist.input* y *namelist.wps* se encuentran en la siguiente ruta:

```
/LUSTRE/ID/hidromet/WRF/namelist_pronosticos/4.2.1.
```

No olvidar verificar que la fecha de *namelist.wps* sea la fecha requerida para el experimento y para sólo tres horas de ejecución en modo pronóstico para este ejemplo

En este caso se correrá el día 12 de Junio de 2022.

Verificar la configuración del dominio propuesto:

```
# Verificación de Dominios propuestos:
cp /LUSTRE/ID/hidromet/WRF/WRF_NCL_scripts/plotgrids_new.ncl .
ml purge
ml load herramientas/ncl/6.6.2
ncl plotgrids_new.ncl
# Revisar desde Jupyter la creación del archivo PDF
```

A continuación se muestra, un [ejemplo](#) de la etapa de preprocesamiento con WPS:

Running WPS

- You are now ready to begin running WPS and WRF. Start by going to the WPS directory:

```
cd WPS
```

- Make any changes to the `namelist.wps` file, to reflect information for your particular run
- Before running `geogrid`, make sure that you have your `geog_data_path` set to the location where you put your geography static data. Once that is set, you can run `geogrid`.

```
./geogrid.exe >& log.geogrid
```

- If you successfully created a `geo_em*` file for each domain, then you are ready to prepare to run `ungrib`. Start by linking in the input GFS data:

```
./link_grib.csh path_where_you_placed_GFS_files
```

Then link to the correct Vtable (GFS, for this case):

```
ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable
```

Then run the `ungrib` executable:

```
./ungrib.exe
```

You should now have files with the prefix "FILE" (or if you named them something else, they should have that prefix)

- You are now ready to run `metgrid`:

```
./metgrid.exe >& log.metgrid
```

You should now have files with the prefix `met_em*` for each of the time periods for which you are running.

- Creación de script de ejecución con SLURM para **WPS-etapa geogrid**.

NOTA: No olvidar sustituir el nombre de la partición por la partición seleccionada para correr la prueba, así como otros parámetros.

```
#!/bin/bash
# script run-geogrid.sh
#SBATCH -J demo_WRF4_geogrid
#SBATCH -p id
#SBATCH -N1
#SBATCH --ntasks-per-node 1
#SBATCH -o slurm.%x.%j.out
#SBATCH -e slurm.%x.%j.err
```

```
cd $HOME/WRFV4/WPS
```

```
ml load intel/2022u2
```

```
ml load curl hdf5 jasper libaec libpng mpi netcdf-c netcdf-fortran zlib
```

```
ml load wrf/4.2.1
srun geogrid.exe
```

Después de ejecutar `geogrid.exe` se debieron haber creado los archivos `geo_em.d01.nc` y `geo_em.d02.nc` correspondientes a los dominios 1 y 2 con la configuración del Pronóstico Operativo del Grupo IOA.

Para correr el anterior `script` sólo ejecutar:

```
sbatch run-geogrid.sh

# Preparación ungrib (Archivos GFS* del 2022-06-12):
# Ejecutar el script:
./link_grib.csh /LUSTRE/ID/hidromet/WRF/2022_06_12_00/entradas/GFS/gfs*

# Enseguida correr los script sbatch editando el nombre del trabajo y
# el programa correspondiente a la etapa de WPS.

# Correr ungrib.exe
sbatch run-ungrib.sh

# Finalmente, si todo corrió exitosamente, ejecutar metgrid.exe
sbatch run-metgrid.sh

* LOS ARCHIVOS GFS DESDE EL 2017 SE ENCUENTRAN EN LA SIGUIENTE RUTA:
* /LUSTRE/KRAKEN/DATOS2/Pronosticos/Salidas/GFS*
```

Sesión 3

☐ Ejecutar WRF.

NOTA: No olvidar verificar que la fecha de `namelist.input` sea la requerida y para sólo 1 día de ejecución en modo pronóstico, según los requerimientos de la ejecución.

Para ejecutar el modelo `wrf.exe` es necesario vincular los archivos para la ejecución del modelo WRF, asociados a la versión del modulo de WRF 4.2.1

```
cd ../WRF
ln -s /LUSTRE/ID/hidromet/WRF/wrf_run_files/* .
```

Enseguida se pueden correr los scripts de sbatch `run-real.sh` y `run-wrf.sh`:

```
# Previamente, vincular los archivos met_em.d0* al directorio WRF
# creado:

cd ../WRF
```

```
ln -sf ../WPS/met_em.d0* .
```

```
### SCRIPT run-real.sh
```

```
# Script run-real.sh:
```

```
#!/bin/bash
```

```
# script run-real.sh
```

```
#SBATCH -J real_WRF4
```

```
#SBATCH -p id
```

```
#SBATCH -N 3
```

```
#SBATCH --ntasks-per-node 22
```

```
#SBATCH -o slurm.%x.%j.out
```

```
#SBATCH -e slurm.%x.%j.err
```

```
cd $HOME/WRFV4/WRF
```

```
ml load intel/2022u2/compilers
```

```
ml load curl hdf5 jasper libaec libpng mpi netcdf-c netcdf-fortran zlib
```

```
ml load wrf/4.2.1
```

```
export CORES=66
```

```
/sbin/logsave REGISTRO_REAL.txt mpirun -np $CORES real.exe
```

```
### SCRIPT run-wrf.sh
```

```
# Script run-wrf.sh
```

```
#!/bin/bash
```

```
# script run-wrf.sh
```

```
#SBATCH -J wrf_WRF4
```

```
#SBATCH -p id
```

```
#SBATCH -N 3
```

```
#SBATCH --ntasks-per-node 22
```

```
#SBATCH -o slurm.%x.%j.out
```

```
#SBATCH -e slurm.%x.%j.err
```

```
cd $HOME/WRFV4/WRF
```

```
ml load intel/2022u2/compilers
```

```
ml load curl hdf5 jasper libaec libpng mpi netcdf-c netcdf-fortran zlib
```

```
ml load wrf/4.2.1
```

```
export CORES=66
```

```
/sbin/logsave REGISTRO_WRF.txt mpirun -np $CORES wrf.exe
```

NOTA: Verificar el nombre de la partición, el directorio de trabajo, el número de cores y las directivas de SLURM para asignar el número de nodos y tareas por nodo, según los requerimientos y la partición donde se vaya a correr el modelo.

Una vez que ya se tienen los scripts de SLURM disponibles, ya es posible mandar a correr primero la etapa de *real.exe* y al finalizar *wrf.exe*:

Se mandan a correr los siguientes scripts:

```
sbatch run-real.sh
# Al finalizar la ejecución del real.exe y SOLO SI fue exitosa la ejecución
se debieron de haber creado los archivos con las condiciones de frontera y
las condiciones iniciales. A esta etapa se le conoce como interpolacion
vertical.
Proceder a correr el script de WRF siguiente:
```

```
sbatch run-wrf.sh
```

MONITOREO EJECUCIÓN DE WRF

Se pueden utilizar las siguientes instrucciones para determinar el estado de ejecución de los trabajos enviados al Clúster:

```
# Verificar el estado del trabajo en SLURM:
squeue $USER -i 5

# Verificar el estado, tiempos de escritura y avance de la ejecución:
cd $HOME/WRFV4/WRF
tail -f rsl.out.0000

# Inspeccionar en los archivos de registro qué pudo haber fallado, si es el
caso:
less slurm.err.*
```

```
Wed Oct 23 14:16:54 2024
      JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
     1164079      cigom interact analaura  R        48:24      1 node11
     1164074      cigom interact gaby.res  R       2:42:23      1 node2
     1164067      cigom interact analaura  R       3:52:01      1 node10
     1164080    workq2 real_WRF    a.2035  R         0:10      3 node[31-33]
```

Postprocesamiento de salidas generadas en etapas WPS y WRF:

```
# Para postprocesamiento, por ejemplo graficación de variables,
es recomendable utilizar uno nodo de cómputo dedicado. Para ello, utilizar
las siguientes instrucciones:
```

Iniciar una sesión interactiva

```
srun -p id --pty /bin/bash
```

Permite apartar recursos de cómputo especificando número de núcleos y nodos

```
salloc -w node20 -N1 -c 24 -p id
```

Verificar asignación de nodo:

squeue -u \$USER # Además el prompt de la sesión de bash cambia

```
encit00@ometeotl:~$ srun -p id --pty /bin/bash
encit00@node17:~$ squeue -u $USER
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
1324368	id	bash	encit00	R	0:08	1	node17

```
encit00@node17:~$
```

Complementario: TMUX, SLURM, control de trabajos

- ☐ Crear sesión de trabajo en TMUX
- ☐ Salir y recuperar sesión de trabajo en TMUX

Uos de SSH y rsync

- ☐ Ver Documentacion en el directorio faq de Ometeotl: `/LUSTRE/ID/hidromet/faq/cluster/`

MPI

- ☐ Trabajos en ejecución en modo serial y paralelo
- ☐ Ejecución en Paralelo: *memoria compartida y memoria distribuida*

SLURM

- ☐ Formas de acceder a los recursos del Clúster: Sesión interactiva y scripts *sbatch*

Previo

- ☐ Entorno de ejecución y datos de simulación
- ☐ Ejcutar WRF en Clúster Ometeotl
- ☐ Ejecución vía `srun` y `sbatch`

MATLAB:

Se puede crear el siguiente alias del comando `matlab` en la sesión de trabajo en la terminal:

```
alias matlab="/opt/apps/matlab/2019b/bin/./matlab"
```

Python: Bibliotecas y módulos para el estudio de variables meteorológicas y salidas de WRF:

Instalacion de entornos Python en Sistemas GNU/Linux

1. Ambiente `wrf` (Estudio de las salidas del modelo ARW-WRF que se utiliza para correr los Pronósticos Meteorológicos del ICAyCC).

```
# Secuencia de pasos para la instalacion del entorno de trabajo wrf:
cd

# Descargar el script para la instalacion del Gestor de paquetes miniforge
wget -c https://github.com/conda-forge/miniforge/releases/download/24.11.2-1/Miniforge3-24.11.2-1-Linux-x86_64.sh

# Asignar permiso de ejecucion al script e instalar el software en el
directorio $HOME del usuario en el
# directorio miniforge:
chmod u+x Miniforge3-24.11.2-1-Linux-x86_64.sh
./Miniforge3-24.11.2-1-Linux-x86_64.sh -b -f -p ~/miniforge

# Posterior a la instalacion del gestor de software miniforge se ejecutan
los siguientes pasos:
. ~/miniforge/etc/profile.d/conda.sh
conda activate base
conda create --name wrf --file requirements.txt
```

2. Activación del entorno `wrf`:

```
conda activate wrf

# Se puede agregar al archivo `.bashrc` para precargar el gestor de paquetes
conda y tener disponible el
# entorno wrf cada vez que se inicie sesion en BASH. Para ello agregar al
archivo .bashrc las siguientes
# lineas:

source $HOME/miniforge/etc/profile.d/conda.sh
conda activate wrf
```

3. Declarar un nuevo kernel en Jupyter, con el entorno `wrf` recién creado:

```
# Borrar el archivo ubicado en:
$HOME/miniforge/envs/<ENV_NAME>/share/jupyter/kernels/python3/kernel.json
```

```
# Registrar el nombre "wrf" al kernel:  
python -m ipykernel install --user --name wrf --display-name "Python (wrf)"
```

Github

Desde terminal:

```
# Crear llaves SSH para desarrollo en Github si el repositorio es privado  
  
eval "$(ssh-agent -s)"  
ssh-add .ssh/iaa_github  
  
# Clonar (hacer una copia) del repositorio de github  
git@github.com:paveloropeza/curso_ometeotl.git
```

Referencias:

- https://www2.mmm.ucar.edu/wrf/wrf_tutorial/build/html/index.html
- https://www2.mmm.ucar.edu/wrf/users/tutorial/presentation_pdfs/202307/
- https://www2.mmm.ucar.edu/wrf/users/tutorial/presentation_pdfs/
- <https://registry.opendata.aws/noaa-gfs-bdp-pds/>
- https://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html
- http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/contents.html
- https://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP8
- <https://www2.mmm.ucar.edu/wrf/OnLineTutorial/Basics/>
- <https://docs.nersc.gov/applications/wrf/>
- <https://hpc.llnl.gov/sites/default/files/scancel.txt>
- <https://slurm.schedmd.com/quickstart.html>
- https://lmod.readthedocs.io/en/latest/010_user.html
- <https://www.ocf.berkeley.edu/~ckuehl/tmux/>
- <https://www.hawkhost.com/blog/2010/06/28/tmux-the-terminal-multiplexer/>