



**CCAD**

Centro de Computación  
de Alto Desempeño de la  
Universidad Nacional de Córdoba



UNC

Universidad  
Nacional  
de Córdoba

# Instalación y ejecución de WRF en Mendieta

## Índice

1. Introducción
2. Descarga de WRF/WPS/ARWpost
3. Instalación de WRF + dependencias
4. Obtención de datos terrestres
5. Ejecución del modelo
6. Análisis y control de ejecución
7. Bibliografía & Guías de instalación tomadas de referencia

---

## 1. Introducción

La implementación de WRF en el cluster Mendieta se encuadra en el marco de la Tesis de Licenciatura en Ciencias de la Computación(**FaMAF-UNC**) de Luis Miguel Vargas Calderon.

Código utilizado:

Procesamiento: WRF3.6.1

Pre-procesamiento: WPS3.6.1

Post-procesamiento: ARWpost\_V3

Para versión WRF3.8 Realizar este procedimiento cambiando 3.6.1 por 3.8 #Bajo estudio en este momento.

Requerimientos:

Instalados en Mendieta:

- perl
- netcdf
- hdf5
- openmpi

No instalados en Mendieta:

- Jasper: Herramienta adicional para pre-procesamiento
- Grads: Herramienta adicional para post-procesamiento

---

## 2. Descarga de WRF/WPS/ARWpost

clonar este repo:

```
ssh <USER>@mendieta.ccad.unc.edu.ar
cd $HOME
git clone https://github.com/lvc0107/wrf_mendieta.git
cd wrf_mendieta
mkdir WRF3.6.1
```

Cargar las siguientes variables de entorno

```
. set_configuration.sh
```

## Descarga de WRF

```
cd $WRF_BASE/WRF3.6.1
wget http://www2.mmm.ucar.edu/wrf/src/WRFV3.6.1.TAR.gz
tar -xvzf WRFV3.6.1.TAR.gz
rm WRFV3.6.1.TAR.gz
```

## Descarga de WPS

```
cd $WRF_BASE/WRF3.6.1
wget http://www2.mmm.ucar.edu/wrf/src/WPSV3.6.1.TAR.gz
tar -xvzf WPSV3.6.1.TAR.gz
rm WPSV3.6.1.TAR.gz
```

## Descarga de ARWpost

```
cd $WRF_BASE/WRF3.6.1
wget http://www2.mmm.ucar.edu/wrf/src/ARWpost_V3.tar.gz
tar -xvzf ARWpost_V3.tar.gz
rm ARWpost_V3.tar.gz
```

---

## 3. Instalación de WRF + dependencias

### 3.1. Seteo de entorno

Jasper:

```
cd $WRF_BASE

module load compilers/gcc/4.9
mkdir -p library/jasper
cd library/jasper
wget http://www.ece.uvic.ca/~mdadams/jasper/software/jasper-1.900.1.zip
unzip jasper-1.900.1.zip
cd jasper-1.900.1
./configure --prefix=$WRF_BASE/library/jasper
make
make check
make install
```

Chequeo de la correcta instalación de jasper:

```
ls ../bin/
imgcmp  imginfo  jasper  tmrdemo
```

**ATENCIÓN!!!** La siguiente sección debe usarse en caso de que las dependencias de MENDIETA no estén instaladas.

Actualmente las dependencias necesarias si están instaladas por lo tanto pasamos directamente a la sección 3.1.2.

En caso de que no estuviesen instaladas seguir en la siguiente sección. También es importante cambiar “set\_configuration.sh” por “set\_custom\_configuration.sh” en el archivo run\_wrf\_model.sh.

### 3.1.1 Instalación de tools propias (Sin usar las que provee Mendieta)

Cargar las siguientes variables de entorno

```
. set_custom_configuration.sh
```

Zlib

```
cd $WRF_BASE/library
mkdir zlib
cd zlib
wget http://fossies.org/linux/misc/zlib-1.2.8.tar.gz
tar -xvf zlib-1.2.8.tar.gz
rm zlib-1.2.8.tar.gz
cd zlib-1.2.8/
./configure --prefix=$(pwd)
make test
make install
```

HDF5

```
cd $WRF_BASE/library
mkdir hdf5
cd hdf5/
wget https://www.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8.13/src/hdf5-1.8.13.tar.gz
tar -xvzf hdf5-1.8.13.tar.gz
hdf5-1.8.13.tar.gz
cd hdf5-1.8.13/
./configure --prefix=$(pwd)
make test
make install
make check-install
```

## NETCDF

```
cd $WRF_BASE/library
mkdir netcdf
wget http://pkgs.fedoraproject.org/repo/pkgs/netcdf/netcdf-4.3.3.1.tar.gz/5c9dad3705a3408d27f696e5b31fb88c/netcdf-4.3.3.1.tar.gz
md5sum netcdf-4.3.3.1.tar.gz | grep 5c9dad3705a3408d27f696e5b31fb88c
tar -xvf netcdf-4.3.3.1.tar.gz
rm netcdf-4.3.3.1.tar.gz
cd netcdf-4.3.3.1/
./configure --prefix=$(pwd)/.. FC=gfortran F77=gfortran CC=gcc --enable-shared LDFLAGS="-L$HOME/WRF/library/hdf5/hdf5-1.8.13/lib" CPPFLAGS="-I$HOME/WRF/library/hdf5/hdf5-1.8.13/include"
make
make check
make install
```

## NETCDF-Fortran

```
cd $WRF_BASE/library/netcdf
wget wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-fortran-4.2.tar.gz
tar -xvf netcdf-fortran-4.2.tar.gz
rm netcdf-fortran-4.2.tar.gz
cd netcdf-fortran-4.2
./configure --prefix=$(pwd)/.. FC=gfortran F77=gfortran CC=gcc --enable-shared 2>&1 | tee configure.log
make
make check
make install
```

## MVAPICH

```
cd $WRF_BASE/library
mkdir mvapich
cd mvapich
wget http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.2.tar.gz
tar -xvf mvapich2-2.2.tar.gz
rm mvapich2-2.2.tar.gz
cd mvapich2-2.2
#configure: error: 'infiniband/mad.h not found. Please retry with --disable-mcast'
./configure --prefix=$(pwd)/.. --disable-mcast
make
make install

# Add $(pwd)/../bin to PATH
```

### 3.1.2 Uso de tools instaladas en Mendieta

```
. set_configuration.sh
```

### 3.2. Instalación de WRF

```
cd $WRF_DIR
./clean -a
./configure
```

Al iniciar configure debe dar un mensaje como el siguiente:

De esta pinta si se esta usando set\_configuration.sh (Herramientas provistas por Mendieta. RECOMENDADO)

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /opt/netcdf-fortran/4.4.2-netcdf_4.3.3.1-gcc_4.9.2
Will use PHDF5 in dir: /opt/hdf5/1.8.15-gcc_4.9.2
which: no timex in (/opt/netcdf-fortran/4.4.2-netcdf_4.3.3.1-gcc_4.9.2/bin:/opt/netcdf/4.3.3.1-gcc_4.9.2/bin:/opt/hdf5/1.8.15-gcc_4.9.2/bin:/opt/openmpi-cuda/1.8.8-gcc_4.9-cuda_7.0-clean/bin:/opt/gcc/4.9.3/bin:/opt/cuda/7.0/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/ibutils/bin:/opt/mendieta/bin:/home/alighezzolo/bin:/home/alighezzolo/conae/library/grads-2.0.2/bin)
```

O de esta pinta si se esta usando set\_custom\_configuration.sh

```
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/<USER>/wrf_mendieta/library/netCDF
Will use PHDF5 in dir: /home/<USER>/wrf_mendieta/library/hdf5-1.8.13
which: no timex in (/opt/gcc/4.9.3/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/ibutils/bin:/opt/mendieta/bin:/home/<USER>/bin)
```

Verificar que las variables **NETCDF** y **PHDF5** apunten a los path seteados en los archivos set\_configuration.sh (set\_custom\_configuration.sh).

Elegir opciones 34-1 para usar procesos con memoria distribuida: openmpi

```
34. x86_64 Linux, gfortran compiler with gcc (dmpar)
Compile for nesting? (1=basic) 1
```

Para utilizar openmpi, se debe actualizar la variable DM\_CC con el valor -DMPI2\_SUPPORT en el archivo configure.wrf.

```
DM_CC          =      mpicc -DMPI2_SUPPORT
```

En caso de correr WRF en un solo nodo es necesario usar procesos con memoria compartida: openMP.

```
33. x86_64 Linux, gfortran compiler with gcc (smpar)
Compile for nesting? (1=basic) 1
```

```
./compile em_real &> compile.log
```

Comprobar la generación de los siguientes archivos .exe:

```
ls -lt main/*.exe

real.exe
tc.exe
nup.exe
ndown.exe
wrf.exe
```

### 3.3 Instalación de WPS

```
cd $WPS_DIR
./clean -a
./configure
```

Notar que al iniciar debe dar un mensaje como el siguiente:

```
Will use NETCDF in dir: /home/<USER>/library/netCDF
Found Jasper environment variables for GRIB2 support...
$JASPERLIB = /home/<USER>/wrf_mendieta/library/jasper/lib
$JASPERINC = /home/<USER>/wrf_mendieta/library/jasper/include
```

Elegir opción 1

Actualizar Vtable.

```
cd ungrib/Variable_Tables
wget http://www2.mmm.ucar.edu/wrf/src/Vtable.GFS_new
cd ../../
ln -s ungrib/Variable_Tables/Vtable.GFS_new Vtable
```

En caso de correr WRF en un solo nodo es necesario se debe declarar explícitamente el uso de openMP agregando el flag **-lgomp** a la variable WRF\_LIB en el archivo configure.wps.

```
WRF_LIB      =      -L$(WRF_DIR)/external/io_grib1 -lio_grib1 \
                    -L$(WRF_DIR)/external/io_grib_share -lio_grib_share \
                    -L$(WRF_DIR)/external/io_int -lwrfio_int \
                    -L$(WRF_DIR)/external/io_netcdf -lwrfio_nf \
                    -L$(NETCDF)/lib -lnetcdf -lnetcdf -lgomp
```

```
./compile &> compile.log
```

Comprobar la generación de los siguientes archivos .exe:

```
ls -lt *.exe
```

```
metgrid.exe -> metgrid/src/metgrid.exe
ungrib.exe -> ungrib/src/ungrib.exe
geogrid.exe -> geogrid/src/geogrid.exe
```

Copiar este script:

```
cp $WRF_BASE/link_grib.csh $WPS_DIR
```

### 3.4 Instalación de ARWpost

```
cd $ARWPOST_DIR
```

Agregar -lnetcdf en src/Makefile

```
ARWpost.exe: $(OBJJS)
    $(FC) $(FFLAGS) $(LDFFLAGS) -o $@ $(OBJJS)
    -L$(NETCDF)/lib -I$(NETCDF)/include -lnetcdf -lnetcdf
```

```
./clean -a
./configure
```

Elegir opción 3.

```
./compile
```

Comprobar la generación del siguiente archivo .exe:

```
ls *.exe
ARWpost.exe
```

### 3.5 Instalación de grads

```
cd $WRF_BASE/library
wget ftp://cola.gmu.edu/grads/2.0/grads-2.0.2-bin-CentOS5.8-x86_64.tar.gz
tar -xvzf grads-2.0.2-bin-CentOS5.8-x86_64.tar.gz
rm grads-2.0.2-bin-CentOS5.8-x86_64.tar.gz
cd grads-2.0.2
mkdir data
```



---

#### 4. Obtención de datos terrestres

```
cd $WPS_DIR
wget http://www2.mmm.ucar.edu/wrf/src/wps_files/geog_complete.tar.bz2
tar -xjvf geog_complete.tar.bz2
rm geog_complete.tar.bz2

#datos adicionales
cd geog
wget http://www2.mmm.ucar.edu/wrf/src/wps_files/topo_gmted2010_30s.tar.bz2
tar -xjvf topo_gmted2010_30s.tar
rm topo_gmted2010_30s.tar

wget http://www2.mmm.ucar.edu/wrf/src/wps_files/topo_30s.tar.bz2
tar -xjvf topo_30s.tar.bz2
rm topo_30s.tar.bz2

wget http://www2.mmm.ucar.edu/wrf/src/wps_files/modis_landuse_21class_30s.tar.bz2
tar -xjvf modis_landuse_21class_30s.tar.bz2
rm modis_landuse_21class_30s.tar.bz2
```

## 5. Ejecución del modelo

Configuración de entorno:

```
cd $WRF_BASE/  
. set_configuration.sh  
chmod +x run_wrf_model.*      # Solo una vez es necesario  
mkdir gribfiles
```

Actualizar namelist.wps con path al directorio geog creado en el step anterior

```
cd $WRF_BASE/scenarios  
#Edit namelist.wps  
geog_data_path = '/home/<USER>/wrf_mendieta/<WRF_VERSION>/WPS/geog' # <USER> y <WRF_VERSION>  
> que correspondan
```

### 5.1. Crear el directorio escenarios con la siguiente estructura:

```
tree scenarios  
scenarios  
├─ Scenario1  
│   ├── namelist.ARWpost  
│   └─ namelist.input  
├─ Scenario2  
│   ├── namelist.ARWpost  
│   └─ namelist.input  
├─ Scenario3  
│   ├── namelist.ARWpost  
│   └─ namelist.input  
│   .  
│   .  
│   .  
├─ ScenarioN  
│   ├── namelist.ARWpost  
│   └─ namelist.input  
├─ gradfile1.gs  
├─ gradfile2.gs  
├─ .  
├─ .  
├─ .  
├─ gradfileN.gs  
└─ namelist.wps
```

Los archivos **namelist.{wps, input, arwpost}** creados en la estructura de directorios anterior son inputs de configuración necesarios para cada una de las siguientes etapas:

pre-procesamiento: utiliza namelist.wps

procesamiento: utiliza namelist.wrf

post-procesamiento: utiliza namelist.arwpost

Tal como han sido creado en la estructura de directorios anterior funcionan como templates.

Se deben configurar cada vez que se considere necesario, pero dejándolos siempre dentro del directorio escenarios.

El script que lanza los jobs genera una copia de estos templates, les actualiza las fechas y los deploya en los directorios necesarios para que WRF los procese.

## Ejemplo usado para CAEARTE

```
tree scenarios
scenarios
├─ A_Thompson_MYJ
│   ├── namelist.ARWpost
│   └─ namelist.input
├─ B_Marrison_MYJ_sf_sfclay_physics
│   ├── namelist.ARWpost
│   └─ namelist.input
├─ cbar.gs
├─ C_WDM6_QNSE_sf_sfclay_physics
│   ├── namelist.ARWpost
│   └─ namelist.input
├─ D_WRF6_MYJ_sf_sfclay_physics
│   ├── namelist.ARWpost
│   └─ namelist.input
├─ E_WDM6_MYNN3
│   ├── namelist.ARWpost
│   └─ namelist.input
├─ HPC_CBA_Rain.gs
├─ HPC_CBA_Tmax_Min.gs
├─ meteogramas_Precipitation.gs
├─ meteogramas_rh.gs
├─ meteogramas_Temp.gs
├─ meteogramas_WindDir.gs
├─ meteogramas_WindSpeed.gs
├─ namelist.wps
└─ rgbset.gs
```

## 5.2 Archivos configurables por el usuario

1) namelist.wps: configuración para etapa de pre-prosesamiento.

Las fechas son actualizadas automáticamente por el script run\_wrf\_model.py

```
cd $WRF_BASE/scenarios
cat namelist.wps

&share
  wrf_core = 'ARW',
  max_dom = 1,
  start_date = 2016-10-20_00:00:00
  end_date = 2016-10-21_12:00:00
  interval_seconds = 10800
  io_form_geogrid = 2,
/

&geogrid
  parent_id      = 1, 1,
  parent_grid_ratio = 1, 3,
  i_parent_start = 1, 37,
  j_parent_start = 1, 83,
  e_we          = 300, 61,
  e_sn          = 250, 91,
  geog_data_res  = '30s', '30s',
  dx = 4000,
  dy = 4000,
  map_proj = 'lambert',
  ref_lat  = -32.4,
  ref_lon  = -66.0,
  truelat1 = -60.0,
  truelat2 = -30.0,
  stand_lon = -63.6,
  geog_data_path = '/home/lvargas/wrf_mendieta/WRF.3.6.1/WPS/geog'
/

&ungrib
  out_format = 'WPS',
  prefix = 'GFS25',
/

&metgrid
  fg_name = 'GFS25'
  io_form_metgrid = 2,
/
```

2) scenariioi/namelist.input: Configuración para etapa de procesamiento. (Para cada escenario)

Las fechas son actualizadas automáticamente por el script run\_wrf\_model.py

```

cd $WRF_BASE/scenarios
cat scenarios/A_Thompson_MYJ/namelist.input

&time_control
run_days              = 0
run_hours             = 36
run_minutes           = 0
run_seconds           = 0
start_year            = 2016
start_month           = 10
start_day             = 20
start_hour            = 00
start_minute          = 00
start_second          = 00
end_year              = 2016
end_month             = 10
end_day               = 21
end_hour              = 12
end_minute            = 00
end_second            = 00
interval_seconds      = 10800
input_from_file       = .true.,.false.,.true.,
history_interval      = 60, 60, 60,
frames_per_outfile    = 1000, 1000, 1000,
restart               = .false.,
restart_interval      = 5000,
io_form_history       = 2
io_form_restart       = 2
io_form_input         = 2
io_form_boundary      = 2
debug_level           = 0
/

&domains
time_step             = 15,
time_step_fract_num   = 0,
time_step_fract_den   = 1,
max_dom               = 1,
e_we                  = 300, 61,
e_sn                   = 250, 91,
e_vert                = 35, 28, 28,
p_top_requested       = 5000,
num_metgrid_levels    = 32,
num_metgrid_soil_levels = 4,
dx                    = 4000, 10000, 3333.33,
dy                    = 4000, 10000, 3333.33,
grid_id               = 1, 2, 3,
parent_id             = 1, 1, 2,
i_parent_start        = 1, 37, 30,
j_parent_start        = 1, 83, 30,
parent_grid_ratio      = 1, 3, 3,
parent_time_step_ratio = 1, 3, 3,
feedback              = 1,
smooth_option         = 0
/

&physics
mp_physics            = 8, 2, 2,
ra_lw_physics         = 1, 1, 1,
ra_sw_physics         = 2, 1, 1,

```

```

radt                = 4,    30,    30,
sf_sfclay_physics   = 2,    1,    1,
sf_surface_physics  = 2,    2,    2,
bl_pbl_physics      = 2,    1,    1,
bldt                = 0,    0,    0,
cu_physics           = 0,    5,    0,
cudt                = 5,    5,    5,
isfflx              = 1,
ifsnow               = 1,
icloud               = 1,
surface_input_source = 1,
num_soil_layers      = 4,
sf_urban_physics     = 0,    0,    0,
/

&fdda
/

&dynamics
w_damping            = 0,
diff_opt              = 1,
km_opt                = 4,
diff_6th_opt          = 0,    0,    0,
diff_6th_factor       = 0.12, 0.12, 0.12,
base_temp              = 290.
damp_opt              = 0,
zdamp                 = 5000., 5000., 5000.,
dampcoef              = 0.2, 0.2, 0.2,
khdif                 = 0,    0,    0,
kvdif                 = 0,    0,    0,
non_hydrostatic       = .true., .true., .true.,
moist_adv_opt          = 1,    1,    1,
scalar_adv_opt         = 1,    1,    1,
/

&bdy_control
spec_bdy_width        = 5,
spec_zone              = 1,
relax_zone             = 4,
specified              = .true., .false., .false.,
nested                 = .false., .true., .true.,
/

&grib2
/

```

3) scenarioid/namelist.ARWpost: Configuración para etapa de post-procesamiento. (Para cada escenario)

Las fechas son actualizadas automáticamente por el script run\_wrf\_model.py

```
cd $WRF_BASE/scenarios
cat escenarios/A_Thompson_MYJ/namelist.ARWpost

&datetime
  start_date = 2016-10-20_00:00:00
  end_date = 2016-10-21_12:00:00
  interval_seconds = 3600,
  tacc = 0,
  debug_level = 0,
/

&io
  input_root_name = '../wrf_run/wrfout_d01_2016-10-20_00:00:00',
  output_root_name = './output/output'
  plot = 'all_list'
  fields = 'height,pressure,tk,tc,rh2,wd10,ws10'
  mercator_defs = .true.
/
  split_output = .true.
  frames_per_outfile = 2

  plot = 'all'
  plot = 'list'
  plot = 'all_list'
! Below is a list of all available diagnostics
  fields = 'height,geopt,theta,tc,tk,td,td2,rh,rh2,umet,vmet,pressure,u10m,v10m,wdir,wspd,wd
10,ws10,slp,mcape,mcin,lcl,lfc,cape,cin,dbz,max_dbz,clfr'

&interp
  interp_method = 0,
  interp_levels = 1000.,950.,900.,850.,800.,750.,700.,650.,600.,550.,500.,450.,400.,350.,300.,25
0.,200.,150.,100.,
/
  extrapolate = .true.

  interp_method = 0,      ! 0 is model levels, -1 is nice height levels, 1 is user specified
  pressure/height

  interp_levels = 1000.,950.,900.,850.,800.,750.,700.,650.,600.,550.,500.,450.,400.,350.,300.,25
0.,200.,150.,100.,
  interp_levels = 0.25, 0.50, 0.75, 1.00, 2.00, 3.00, 4.00, 5.00, 6.00, 7.00, 8.00, 9.00, 1
0.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,
```

4) set\_configuration.sh : Este archivo carga los módulos a usar: compilador , MPI, etc. por default usa gcc y openmpi Pero eventualmente se podrían usar otras opciones como mvapich e icc. Sin embargo si se desea probar otros módulos deben

Compilarse todas las soluciones de nuevo, ie modificar el archivo set\_configuration.sh y volver a realizar todos los pasos desde el paso 1: Instalación de WRF y dependencias

5) set\_custom\_configuration.sh: Idem anterior

Ver módulos que carga set\_configuration.sh

```
module list
Currently Loaded Modulefiles:
  1) cuda/7.0
  2) libs/gcc/4.9
  3) compilers/gcc/4.9
  4) mpi/openmpi/1.8.4-gcc_4.9.2
  5) libs/hdf5/1.8.15-gcc_4.9.2
  6) libs/netcdf/4.3.3.1-gcc_4.9.2
  7) libs/netcdf-fortran/4.4.2-netcdf_4.3.3.1-gcc_4.9.2
```

Ver todos los módulos disponibles en el cluster

```
module avail
```

### 5.3 Correr script: run\_wrf\_model.py

Este script realiza las siguientes tareas:

- 1) Descarga grib files dada una fecha en el directorio gribfiles creado en el step anterior
- 2) Actualiza fecha en namelist.wps en el directorio escenarios
- 3) Actualiza fecha en los namelist.input dentro de cada directorio escenarios/Scenarioi con i:{1..N}
- 4) Actualiza fecha en namelist.ARWpost dentro de cada directorio escenarios/Scenarioi con i:{1..N}
- 5) Ejecuta el modelo para cada uno de los escenarios

```
./run_wrf_model.py --start_date=STARTDATE --offset=OFFSET --nodes=2
```

El script ejecuta todos los escenarios en paralelo corriendo WRF en 2 nodos de la partición capability(40 cores en total).

Ejemplo: Para ejecutar todos los escenarios en dos nodos de Capability (20 cores p/nodo)

```
./run_wrf_model.py --start_date=2016200000 --offset=36 --nodes=2
```

Nota:

Ajustar el tiempo de ejecución del modelo en el script job\_wrf\_N\_nodes.sh de la forma más precisa posible. # Con N en {2, 3, 4, 5, 6, 7, 8}

Ejemplo si la ejecución del modelo toma aproximadamente (poco menos que) una hora y media:

```
SBATCH --time 0-1:30
```



El output de la ejecución es el siguiente:

```
--          -----
\ \      / /  _ \ |  _ _ _ _ |
\ \  / \ / / | |_) | | _ _
\ \ / \ / / | _ / | _ _ |
\ \ / \ / | | \ \ | |
\ \ / \ | | \ \ | |
\ \ / \ | | \ \ | |
```

Start forecast date: 2016-10-20\_00:00:00

End forecast date: 2016-10-21\_12:00:00

```
=====
sbatch job_wrf_2_nodes.sh D_WRF6_MYJ_sf_sfclay_physics 2016-10-20_00:00:00 2016-10-21_12:00:00
Submitted batch job 50360
=====
sbatch job_wrf_2_nodes.sh A_Thompson_MYJ 2016-10-20_00:00:00 2016-10-21_12:00:00
Submitted batch job 50361
=====
sbatch job_wrf_2_nodes.sh C_WDM6_QNSE_sf_sfclay_physics 2016-10-20_00:00:00 2016-10-21_12:00:00
Submitted batch job 50362
=====
sbatch job_wrf_2_nodes.sh E_WDM6_MYNN3 2016-10-20_00:00:00 2016-10-21_12:00:00
Submitted batch job 50363
=====
sbatch job_wrf_2_nodes.sh B_Marrison_MYJ_sf_sfclay_physics 2016-10-20_00:00:00 2016-10-21_12:00:00
Submitted batch job 50364
squeue -u $USER
PARTITION  JOBID  PRIO      NAME      USER ST      TIME NO CPU  GRES NODELIST(REASON)
capability 50360 2472      WRF alighezz R      0:13  2 40 (null mendieta[17-18])
capability 50361 2472      WRF alighezz R      0:13  2 40 (null mendieta[20-21])
capability 50362 2472      WRF alighezz PD     0:00  2 40 (null (Resources))
capability 50363 2472      WRF alighezz PD     0:00  2 40 (null (Resources))
capability 50364 2472      WRF alighezz PD     0:00  2 40 (null (Resources))
```

El script `run_wrf_model.py` ejecuta el comando **squeue -u \$USER** luego de hacer submit de los jobs (ejecución del escenario). Estos jobs están en estado PD (pending) de obtener recursos. Cuando haya nodos disponibles para la ejecución los jobs que obtengan recursos van a pasar a estado R (running).

EL log proporciona también información relevante:

- PARTITION: Partición a la que pertenecen los nodos
- JOBID: Identificador único del job (ejecución del escenario)
- USER: Usuario que lanzo la ejecución
- NAME: Nombre e identificador del job
- TIME: Cuando el job está en estado R este valor se actualiza mostrando el tiempo transcurrido de ejecución. Importante: si el tiempo de ejecución es mayor al estimado en **SBATCH -time** el job se cancela. Por lo tanto es necesario actualizar ese valor en el script `job_wrf_N_nodes.sh` de manera que ese valor sea mayor y correr nuevamente.
- NO: números de nodos asignados
- CPU: número de cores asignados

- NODELIST: lista de nodos asignados al job

Para ejecutar solo un escenario (por ejemplo A\_Thompson\_MYJ) en dos nodos de Capability (20 cores p/nodo) para la misma fecha de inicio y periodo de 36 hs

```
sbatch job_wrf_2_nodes.sh A_Thompson_MYJ 2016-10-20_00:00:00 2016-10-21_12:00:00
```

La ejecución genera los output en los directorios:

```
$WRF_BASE/output/$RUN_PARAMETERS/meteogramas
```

La ejecución genera logs en los directorios:

```
$WRF_BASE/logs/$RUN_PARAMETERS/$SLURM_JOB_ID
```

Donde RUN\_PARAMETERS está definido en el script job\_wrf\_N\_nodes.sh

# con N en {2, 3, 4, 5, 6, 7, 8}

Ver outputs generados:

```
cd $WRF_BASE
ls -l output/40_cores_A_Thompson_MYJ/meteogramas/
total 3.1M
-rw-rw-r-- 1 alighezzolo alighezzolo 16K Nov 5 06:13 temp_max_A.png
-rw-rw-r-- 1 alighezzolo alighezzolo 398K Nov 5 06:13 temp_max_A.tif
-rw-rw-r-- 1 alighezzolo alighezzolo 15K Nov 5 06:13 temp_min_A.png
-rw-rw-r-- 1 alighezzolo alighezzolo 398K Nov 5 06:13 temp_min_A.tif
-rw-rw-r-- 1 alighezzolo alighezzolo 26K Nov 5 06:13 rain24h_A.png
-rw-rw-r-- 1 alighezzolo alighezzolo 398K Nov 5 06:13 rain24h_A.tif
-rw-rw-r-- 1 alighezzolo alighezzolo 367 Nov 5 06:13 rain_COLONIA_CAROYA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 312 Nov 5 06:13 rain_CAPILLA_DEL_MONTE_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 314 Nov 5 06:13 rain_CANALS_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 341 Nov 5 06:13 rain_BRINCKMANN_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 326 Nov 5 06:13 rain_BIALET_MASSE_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 344 Nov 5 06:13 rain_BERROTARAN_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 338 Nov 5 06:13 rain_BALNEARIA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 314 Nov 5 06:13 rain_ARROYO_CABRAL_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 341 Nov 5 06:13 rain_ARROYITO_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 324 Nov 5 06:13 rain_ALTA_GRACIA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 368 Nov 5 06:13 rain_ALMAFUERTE_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 338 Nov 5 06:13 rain_ALICIA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 314 Nov 5 06:13 rain_ALEJO_LEDESMA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 341 Nov 5 06:13 rain_ELENA_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 340 Nov 5 06:13 rain_DEVOTO_A.txt
-rw-rw-r-- 1 alighezzolo alighezzolo 359 Nov 5 06:13 rain_DESPENADEROS_A.txt
```

Se genera un reporte en el archivo meteogramas.html para visualizar estadísticas de todos los escenarios por región.

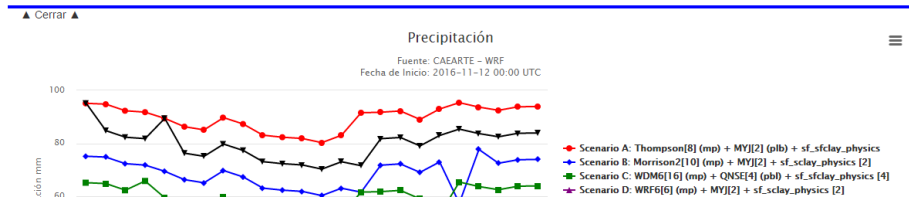
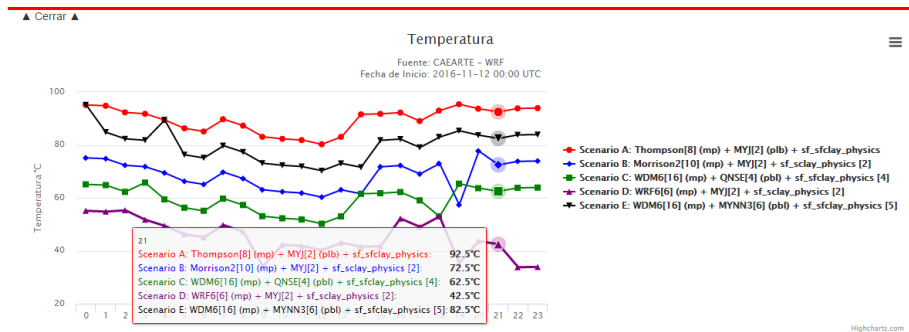
## Meteogramas-WRF

## Selector

Provincia:

Departamento:

Localidad:



También se pueden ejecutar los scripts:

```
job_wrf_1_nodes.sh
job_wrf_3_nodes.sh
job_wrf_4_nodes.sh
job_wrf_5_nodes.sh
job_wrf_6_nodes.sh
job_wrf_7_nodes.sh
job_wrf_8_nodes.sh
```

Ejemplos que ejecutan los escenarios usando 3, 4 y 5 nodos de 20 cores c/u respectivamente

```
./run_wrf_model.py --start_date=2016102000 --offset=36 --nodes=3
./run_wrf_model.py --start_date=2016102000 --offset=36 --nodes=4
./run_wrf_model.py --start_date=2016102000 --offset=36 --nodes=5
```

Importante: La cuota por usuario es de 500GiB. La instalación de WRF ocupa aproximadamente 100GiB (mayormente debido a los ~85 GiB al directorio geog en \$WPS\_DIR)

Por lo tanto quedan disponibles ~400 GiB. Es necesario entonces limpiar (borrar) los resultados que se van generando periódicamente, luego de su procesamiento.

## 6. Análisis y control de ejecución

Durante la ejecución de los jobs podemos ejecutar algunos comandos que nos brindan información del estado de la ejecución:

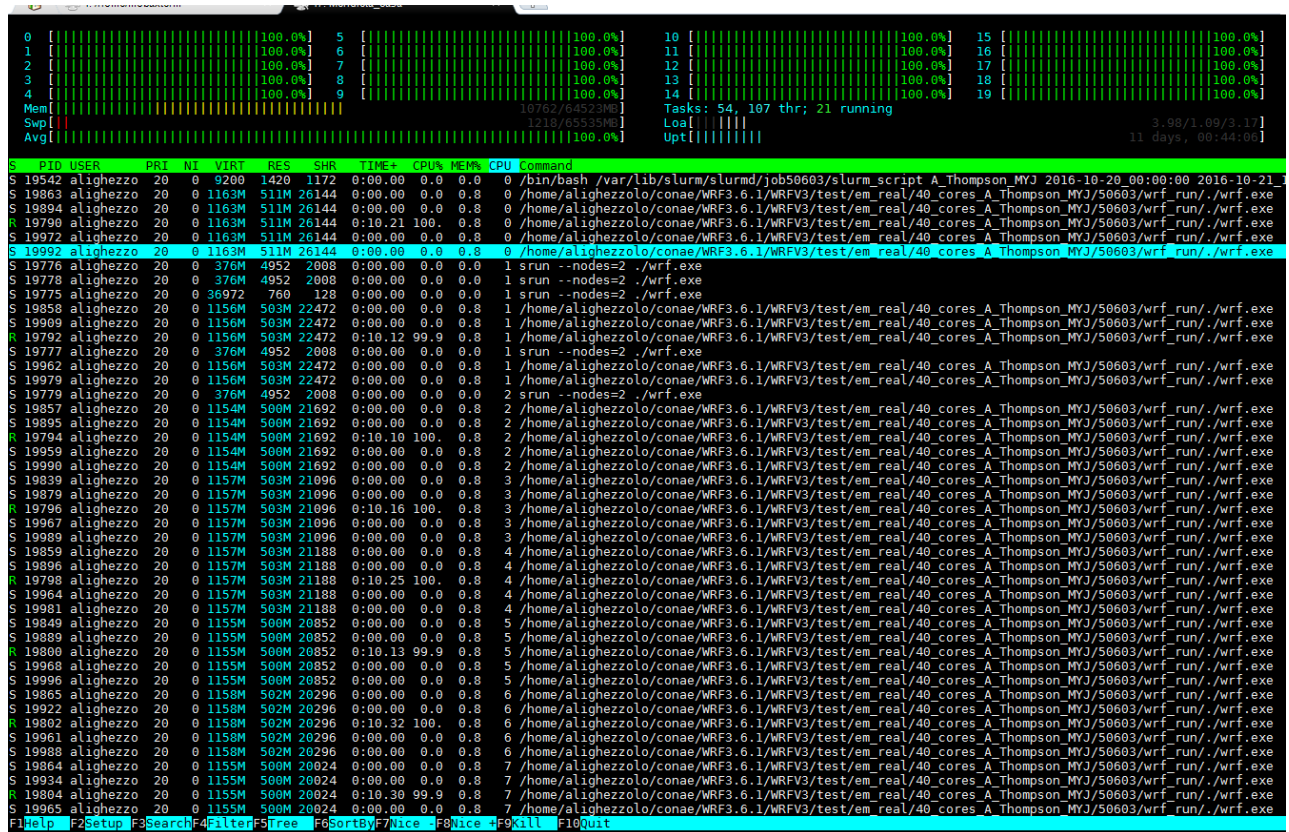
```
squeue -u $USER      # muestra el estado de los jobs propios
squeue               # muestra el estado de todos los jobs en el cluster
```

Si la ejecución de un job está en estado R podemos acceder al nodo para ver la ejecución en tiempo real

Ver uso de los cores en un nodo:

```
squeue -u $USER
capability 50361 2472          WRF alighezz R          0:13 2 40 (null mendieta[20-21])
ssh mendieta20                # también podríamos haber hecho ssh mendieta21

mendieta20 $ htop              # Ver estado de los cores.
```



Ver que funciones de wrf.exe realizan mas computo:

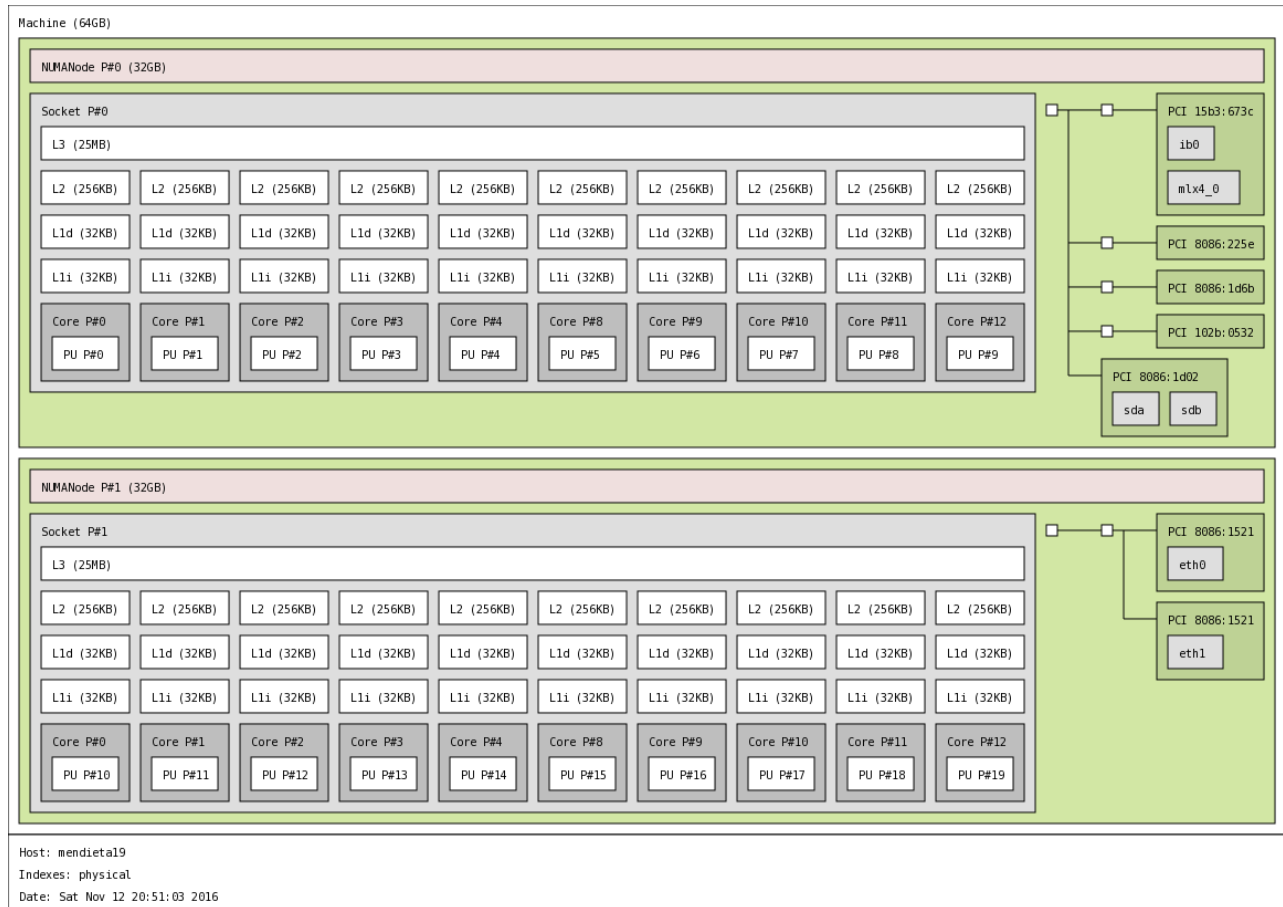
```
ssh mendieta20          # también podríamos haber hecho ssh mendieta21
mendieta20 $ perf top    # Ver funciones que consumen mas computo.
```

```
Samples: 3M of event 'cycles', Event count (approx.): 909350829093
Overhead Shared Object Symbol
12.01% libm-2.12.so [.] _ieee754_powf
10.21% wrf.exe [.] _module_bl_mfshconvpbl_MOD_th_r_from_thl_rt_1d
10.11% libm-2.12.so [.] _ieee754_expj
6.21% wrf.exe [.] _module_advect_em_MOD_advect_scalar_pd
5.97% libm-2.12.so [.] _ieee754_logf
5.37% wrf.exe [.] _module_mp_wdm6_MOD_wdm62d
4.77% [unknown] [k] 0xffffffff81297357
3.68% wrf.exe [.] _module_ra_gsfcsw_MOD_cldflx
3.06% wrf.exe [.] _module_advect_em_MOD_advect_scalar
2.66% wrf.exe [.] _module_bl_mfshconvpbl_MOD_compute_bl89_ml
2.28% wrf.exe [.] _module_bl_mfshconvpbl_MOD_th_r_from_thl_rt_2d
1.50% libm-2.12.so [.] _GI_expj
1.28% wrf.exe [.] _module_em_MOD_rk_update_scalar
1.21% wrf.exe [.] _module_bl_qnsepbl_MOD_qnsepbl
1.02% wrf.exe [.] _module_bl_mfshconvpbl_MOD_mfshconvpbl
0.97% wrf.exe [.] _module_mp_wdm6_MOD_nislfv_rain_plm6
0.95% libm-2.12.so [.] _logf
0.86% wrf.exe [.] _module_small_step_em_MOD_advance_uv
0.78% wrf.exe [.] _module_small_step_em_MOD_advance_w
0.76% wrf.exe [.] _module_big_step_utilities_em_MOD_zero_tend
0.67% wrf.exe [.] _module_big_step_utilities_em_MOD_horizontal_diffusion
0.67% wrf.exe [.] _module_small_step_em_MOD_advance_mu_t
0.66% wrf.exe [.] _module_ra_rrtm_MOD_rtrn
0.63% wrf.exe [.] _module_mp_wdm6_MOD_slope_wdm6
0.60% [unknown] [k] 0xffffffff8152d370
0.48% libm-2.12.so [.] _powf
Press '?' for help on key bindings

Samples: 5M of event 'cycles', Event count (approx.): 979605144611
Overhead Shared Object Symbol
12.22% libm-2.12.so [.] _ieee754_powf
9.51% libm-2.12.so [.] _ieee754_expj
7.91% wrf.exe [.] _module_bl_mfshconvpbl_MOD_th_r_from_thl_rt_1d
6.43% wrf.exe [.] _module_advect_em_MOD_advect_scalar_pd
6.22% wrf.exe [.] _module_mp_wdm6_MOD_wdm62d
6.17% wrf.exe [.] _module_ra_gsfcsw_MOD_cldflx
5.17% libm-2.12.so [.] _ieee754_logf
3.96% [unknown] [k] 0xffffffff81297357
3.07% wrf.exe [.] _module_advect_em_MOD_advect_scalar
2.19% wrf.exe [.] _module_bl_mfshconvpbl_MOD_compute_bl89_ml
1.77% wrf.exe [.] _module_bl_mfshconvpbl_MOD_th_r_from_thl_rt_2d
1.48% libm-2.12.so [.] _GI_expj
1.24% wrf.exe [.] _module_em_MOD_rk_update_scalar
1.19% wrf.exe [.] _module_ra_rrtm_MOD_rtrn
1.15% wrf.exe [.] _module_bl_qnsepbl_MOD_qnsepbl
1.14% wrf.exe [.] _module_mp_wdm6_MOD_nislfv_rain_plm6
0.88% wrf.exe [.] _module_small_step_em_MOD_advance_uv
0.86% wrf.exe [.] _module_bl_mfshconvpbl_MOD_mfshconvpbl
0.83% libm-2.12.so [.] _logf
0.78% wrf.exe [.] _module_small_step_em_MOD_advance_w
0.72% wrf.exe [.] _module_big_step_utilities_em_MOD_zero_tend
0.72% wrf.exe [.] _module_mp_wdm6_MOD_slope_wdm6
0.67% wrf.exe [.] _module_small_step_em_MOD_advance_mu_t
0.66% wrf.exe [.] _module_big_step_utilities_em_MOD_horizontal_diffusion
0.66% [unknown] [k] 0xffffffff8152d370
Press '?' for help on key bindings
[3] 0:ssh*
```

Conocer topología del nodo:

```
ssh mendieta20          # también podríamos haber hecho ssh mendieta21
lstopo                  # Conocer topología del nodo
```



Para hacer pruebas dentro de 1 de capability

```
salloc -p capability -n 20 srun --exclusive --pty --preserve-env $SHELL
salloc: Pending job allocation 38170
salloc: job 38170 queued and waiting for resources
salloc: job 38170 has been allocated resources
salloc: Granted job allocation 38170
[alighezzolo@mendieta11 conae]
[alighezzolo@mendieta11 conae]squeue -u alighezzolo
PARTITION  JOBID  PRIO    NAME    USER  ST      TIME NO CPU  GRES NODELIST(REASON)
capability 38170 2115    srun alighezz R   1:58:54  1  20 (null mendieta11)
[alighezzolo@mendieta11 conae]
```

Ver informacion de la particion capability

```
sinfo -p capability
PARTITION  AVAIL  TIMELIMIT  NODES  STATE NODELIST
capability  up    4-00:00:00    13  alloc mendieta[09-18,20-22]
capability  up    4-00:00:00     1  idle mendieta19
```

## 7. Bibliografía & Guías de instalación tomadas de referencia

- [1] <http://forum.wrfforum.com/viewtopic.php?f=5&t=7099>
- [2] [http://www.unidata.ucar.edu/software/netcdf/docs/building\\_netcdf\\_fortran.html](http://www.unidata.ucar.edu/software/netcdf/docs/building_netcdf_fortran.html)
- [3] [http://www2.mmm.ucar.edu/wrf/users/docs/user\\_guide\\_V3/users\\_guide\\_chap2.htm#\\_Required\\_Compilers\\_and\\_1](http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/users_guide_chap2.htm#_Required_Compilers_and_1)
- [4] [http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation\\_tutorial.php#STEP5](http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP5)
- [5] [http://www2.mmm.ucar.edu/wrf/users/FAQ\\_files/FAQ\\_wrf\\_installation.html](http://www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_installation.html)