

Introducción al Community Earth System Model 2.1.3

27 de marzo de 2023

Ejercicio práctico guiado

Consideraciones generales.

Este ejercicio tiene como objetivo guiar al participante del curso a través del proceso completo de una simulación del *Community Earth System Model*, versión 2.1.3, instalado en el cluster *Leftraru* administrado por el *National Laboratory for High Performance Computing Chile*. Esto incluye la creación de un caso, configuración de la simulación, compilación, corrida, extracción de datos y visualización de resultados.

El ejercicio que realizaremos consiste en una simulación pre-industrial (año 1850 A. D.) con módulos activos (prognósticos) acoplados de tierra (CLM5) y atmósfera (CAM6). A este tipo de simulación se lo conoce como *compset F1850*. Pueden obtener más información sobre este compset (y otros) [aquí](#). Se utilizará una grilla f09_g17, con una resolución horizontal de 0.9° de latitud por 1.25° de longitud. Pueden obtener más información sobre esta y otras grillas [acá](#). Correremos el modelo por un lapso total de tres días, y utilizaremos una frecuencia de guardado de salidas del modelo de seis horas. Registraremos la temperatura superficial (TS) desde CAM6. Pueden consultar sobre variables atmosféricas [aquí](#); y para variables de CLM5 pueden revisar este [link](#).

Los participantes del curso se dividirán en tres grupos, y se compilará y correrá una simulación por grupo. Para esto, dividiremos la sesión online en tres salas, y cada grupo elegirá un miembro del mismo que estará a cargo de correr el modelo, compartiendo pantalla al resto de los miembros del grupo para que todos puedan seguir los pasos del ejercicio con facilidad. Los tres grupos correrán la misma simulación, con una única modificación: cada grupo forzará el modelo con un valor distinto de concentración de dióxido de carbono (CO₂) en la atmósfera. Los valores serán el dado por defecto para el clima pre-industrial, el doble del valor por defecto, y el cuádruple del valor por defecto.

Guía paso a paso.

1) Ingresar al servidor de Leftraru.

Previo al inicio del curso, recibieron un usuario y contraseña para ingresar al servidor. En la terminal, ingresar el siguiente comando:

```
$ ssh -X usuario@lefraru.nlhpc.cl
```

donde deben reemplazar "**usuario**" por el nombre de usuario que han recibido. La opción "-X" se agrega para poder utilizar programas con interfaz gráfica desde el servidor (lo haremos al final del ejercicio). Luego, deberán ingresar la contraseña que les ha sido enviada.

Aclaración: en la línea anterior, "\$" indica el comienzo de la línea, no debe ingresar en la terminal este símbolo (comenzar en la caso anterior con "ssh...")

2) Cargar las librerías y compiladores que necesita CESM

Para ello, ingresar:

```
$ ml use /home/cr2/projects/modulefiles  
$ ml CESMCR2
```

Para corroborar que las dependencias necesarias han sido efectivamente cargadas, ingresar:

```
$ ml
```

Deberían observar los siguientes módulos listados:

```
Currently Loaded Modules:  
1) GCCcore/8.2.0  
2) icc/2019.2.187-GCC-8.2.0-2.31.1  
3) ifort/2019.2.187-GCC-8.2.0-2.31.1  
4) impi/2019.2.187  
5) imkl/2019.2.187  
6) binutils/2.32  
7) intel/2019b  
8) ncurses/6.1  
9) CMake/3.13.4  
10) bzip2/1.0.6  
11) zlib/1.2.11  
12) libreadline/8.0  
13) Tcl/8.6.9  
14) SQLite/3.27.1  
15) GMP/6.1.2  
16) libffi/3.2.1  
17) expat/2.2.6  
18) Perl/5.28.1  
19) XZ/5.2.4  
20) libxml2/2.9.9  
21) XML-LibXML/2.0134-Perl-5.28.1  
22) Szip/2.1.1  
23) HDF5/1.10.4  
24) netCDF/4.6.2  
25) netCDF-Fortran/4.4.4  
26) netCDF-C++4/4.3.0  
27) ESMF/7.1.0r  
28) APR/1.6.5  
29) APR-util/1.6.1  
30) lz4/1.8.3  
31) Serf/1.3.9  
32) utf8proc/2.2.0  
33) Ruby/2.6.2  
34) Java/1.8.0_202  
35) PCRE/8.42  
36) SWIG/3.0.12-Python-2.7.15  
37) Subversion/1.11.1  
38) cURL/7.64.0  
39) gettext/0.19.8.1  
40) git/2.21.0  
41) PyLint/1.9.4-Python-2.7.15  
42) CESM-deps/2  
43) Python/2.7.15  
44) CESMCR2/2.1.3
```

3) Creación de un caso nuevo

Dirigirse al siguiente directorio: `/home/cr2/projects/cesm2/model/cime/scripts/`, utilizando el comando `cd`:

```
$ cd /home/cr2/projects/cesm2/model/cime/scripts/
```

Allí, correr el siguiente comando:

```
$ ./create_newcase --case $CASE --compset $COMPSET --res $GRID --run-unsupported  
!Esto toma ~1 minuto
```

Donde:

- 1) **\$CASE**: corresponde al nombre del caso que correremos, incluyendo la ruta completa al directorio donde se guardará el caso. Ingresar:
`/home/cr2/projects/cesm2/cases/test_co2_XX`, donde "XX" es 01, 02 ó 04, dependiendo de si pertenecen al grupo que utilizará el valor por defecto de la concentración de CO₂ (01), el doble de dicho valor (02), o el cuádruple (04).
- 2) **\$COMPSET**: es el alias del compset que vamos a usar. En nuestro caso: F1850.
- 3) **\$GRID**: la grilla utilizada. En nuestro caso: f09_g17.

La opción `--run-unsupported` es necesaria al utilizar compsets que están definidos, pero no testeados. Si el proceso se completó correctamente (creación del caso), al final debemos leer "Creating case directory `/home/cr2/projects/cesm2/cases/test_co2_XX`".

4) Modificaciones al caso relacionadas a los recursos computacionales

Vayamos ahora a la carpeta donde se creó el caso:

```
$ cd /home/cr2/projects/cesm2/cases/test_co2_XX
```

En esta carpeta, si las librerías las cargamos correctamente, podemos utilizar los comandos "xmlchange" y "xmlquery", para modificar y consultar sobre archivos .xml, los que controlan cómo se compila y ejecuta el modelo. Primero, modificaremos la cantidad de nodos

computacionales que se destinarán a correr el modelo. Leftrarú posee distintos tipos de nodos computacionales. Existen los nodos *slims*, cada uno de los cuales posee 20 núcleos (también llamados cores o CPUs) y los nodos *general*, los cuales poseen 44 núcleos por nodo. Por defecto, la instalación de CESM2 en Leftrarú utiliza nodos *slims*. En este ejercicio en cambio, utilizaremos nodos *general*. En particular, vamos a correr cada simulación con 2 nodos *general* (es decir, con 88 núcleos). Como comparación, una MacBook Pro de última generación posee ocho núcleos). Para hacer estas modificaciones, modificaremos las variables `job_queue`, `costpes_per_node`, `max_mpitasks_per_node`, `max_tasks_per_node`, `ntasks` y `ntasks_esp`.

```
$ ./xmlchange JOB_QUEUE=general !cambia el tipo de nodo
$ ./xmlchange COSTPES_PER_NODE=44 !número de núcleos por nodo
$ ./xmlchange MAX_MPITASKS_PER_NODE=44 !número de núcleos por nodo (tareas MPI)
$ ./xmlchange MAX_TASKS_PER_NODE=44 !máx. # de tareas/hilos permitidos por nodo
$ ./xmlchange NTASKS=-1 !define dos nodos para todos los componentes de CESM2
$ ./xmlchange NTASKS_ESP=1 !define un núcleo para la componente ESP
```

En esta instancia también modificaremos el valor de la concentración de CO₂ atmosférico (para las simulaciones `test_co2_02` y `test_co2_04`). En nuestras simulaciones, vamos a dejar este valor constante (también existe la posibilidad que la concentración de CO₂ sea una variable pronóstica, es decir, que el modelo la prediga). El comando es:

```
$ ./xmlchange CCSM_CO2_PPMV=$VALOR
```

El valor pre-industrial (por defecto, simulación `test_co2_01`) es 284.7 ppmv. Por lo tanto, duplicar o cuadruplicar este valor según corresponda.

5) Setup de la simulación

En el directorio del caso, ejecutaremos el comando `case_setup` para crear los scripts necesarios para correr el modelo:

```
$ ./case.setup
```

 ¡Esto demora <1 minuto

Si se ejecuta bien, deberíamos poder leer al final:

Creating file case.st_archive

Creating user_nl_XXX files for componentes and cpl

If an old case build already exists, might want to run 'case.build -clean' before building

You can now run './preview_run' to get more infor on how your case will be run

6) Cambios en los namelists de cada componente

Una vez seteado el caso, y antes de compilar, modificaremos las variables output que serán guardadas en memoria, así como su frecuencia. Estos cambios los introduciremos en una serie de archivos que se crearon en el paso anterior, los archivos `user_nl_XXX`, donde `XXX` es cada componente, por ejemplo, "cam" es para la componente atmosférica, "clm" para la componente de tierra, etcétera.

Como sólo estamos interesados en guardar la variable "temperatura superficial", y como dicha variable es una variable de la componente atmosférica, sólo debemos modificar `user_nl_cam`. En el directorio del caso, abriremos este archivo utilizando el editor de texto `vim`:

```
$ vim user_nl_cam
```

Primero, eliminaremos todas las variables output de la componente atmosférica que son definidas por defecto, definiendo la variable *empty_htapes* como verdadera. Luego, agregaremos la única variable cuyo output nos interesa guardar (TS), utilizando la variable *fincl1*. Además, definiremos la frecuencia de output de la variable TS cada seis horas usando la variable *nhtfrq* (<0 indica frecuencia en horas; >0 indica frecuencia en días; 0 indica promedio mensual). Finalmente, definiremos la variable *mfilt* como 13, la cual define la cantidad de registros por archivo. Dado que la simulación correrá por tres días, tendremos solo un archivo con los 12 datos (un dato cada seis horas). Pero hay que ingresar $mfilt = 12 + 1 = 13$ (+1 con respecto a la cantidad de datos reales). El archivo *user_nl_cam* debería quedar definido de la siguiente manera:

```
! User should add all user specific namelist changes below in the form of
! namelist_var = new_namelist_value
empty_htapes = .true.
fincl1 = 'TS'
nhtfrq = -6
mfilt = 13
```

De la misma forma, para evitar que el módulo CLM5 registre variables modificaremos el archivo *user_nl_clm* con la siguiente línea:

```
hist_empty_htapes = .true.
```

Nota: la utilización del editor de texto *vim* requiere cierta práctica previa. Seleccionar un miembro del grupo que tenga experiencia utilizándolo.

7) Compilado del caso

Ya estamos listos para compilar el caso. En el directorio del mismo, utilizar el comando *case.build*:

```
$ ./case.build --skip-provenance-check
```

 ¡Esto demora ~30 mins

Si este paso se realizó correctamente, debería aparecer el siguiente mensaje al final: MODEL BUILD HAS FINISHED SUCCESSFULLY.

7) Definición del largo total de la simulación

Una vez compilado el caso, realizar una última modificación antes de enviar el caso a la cola en el directorio del caso: definir el largo total de la simulación como tres días, de la siguiente manera:

```
$ ./xmlchange STOP_N=3
$ ./xmlchange STOP_OPTION=ndays
```

Por defecto, el número definido para *STOP_N* es en términos de días pero de todos modos especificamos que la opción de tiempo (*STOP_OPTION*) es en días. También es posible especificar meses (*nmonths*) o años (*nyears*). .

8) Enviar caso

Desde el directorio del caso, ingresar:

```
$ ./case.submit
```

Si el caso fue enviado a la cola correctamente, aparecerá el mensaje:

```
Submitted job id is xxxxxxxx
```

```
Submitted job case.run with id xxxxxxxx
```

```
Submitted job case.st_archive with id xxxxxxxx
```

9) Visualización de archivos de salida de la simulación

Mientras la simulación está corriendo, los archivos de salida son acumulados en la carpeta *scratch*, específicamente en la ruta *scratch/test_co2_XX/run/*. Es posible revisar el estado de la simulación en esta carpeta.

10) Visualización de resultados finales

Una vez terminada la simulación, los archivos de salida, en formato netcdf (.nc), son transferidos a la carpeta *archives/test_co2_XX/*. Debido a que solo tendremos archivos de CAM6 se encontrarán en el subdirectorio *cam/hist/*. Ya que configuramos *mfilt = 13* y corrimos tres días de simulación, tendremos solo un archivo de salida. Podemos visualizar rápidamente los resultados desde el mismo servidor utilizando el programa *ncview*. Para ello, debemos cargar el programa *ncview*, dirigirse al directorio donde se encuentran los archivos de salida, y utilizar el comando *ncview*:

```
$ ml ncview
```

```
$ cd /home/cr2/projects/cesm2/archives/test_co2_XX/atm/hist
```

```
$ ncview archivo_salida
```

Donde *archivo_salida* contiene los resultados de la simulación.