Adriaclim and CIME

Overview and first steps

F. Viola[1]

[1]CMCC – Centro EuroMediterraneo sui Cambiamenti Climatici
OPA Division – Ocean Predictions and Applications

2021/05/05

Table of contents

Introduction

CIME

Setting up the environment

Workflow

Homework

Adriaclim

Adriaclim is an EU interreg project between Italy and Croatia aiming at:

- enhancing CC adaptation capacity in coastal areas developing homogeneous and comparable data
- improving knowledge, capacity and cooperation on climate change observing and modeling systems
- developing advanced information system, tools and indicators for optimal CC adaptation planning

Our task

Among the many tasks assigned to OPA, one that is particularly relevant is the creation of a coupled model through the CIME platform developed by NCAR. This activity, carried out together with CMCC's REMHI division, is currently focused on the setup of a coupled model between an oceanographic component (i.e., NEMO) and an atmospheric one (i.e., WRF). In the next months, other models will be coupled...

Table of contents

Introduction

CIME

Setting up the environment

Workflow

Homework

Introduction

CIME contains:

- the support scripts (configure, build, run, test)
- data models
- essential utility libraries
- a main
- other tools

to build a **single-executable coupled Earth System Model**. CIME is available in a stand-alone package that can be compiled and tested without active prognostic components but is typically included in the source of a climate model. CIME does not contain: any active components, any intra-component coupling capability (such as atmosphere physics-dynamics coupling).

Table of contents

Introduction

CIME

Setting up the environment

Workflow

Homework

Overview

Let's suppose we want to run CIME on Zeus. We need:

NOTE: we are skipping CIME dependencies, since they are already satisfied on Zeus. Just remember to enable conda for python 2.7.

Overview

Let's suppose we want to run CIME on Zeus. We need:

- CESM

```
$ git clone -b cesm2.1-nemobranch https://github.com/ESCOMP/\
  cesm.git my_cesm2
```

NOTE: we are skipping CIME dependencies, since they are already satisfied on Zeus. Just remember to enable conda for python 2.7.

Overview

Let's suppose we want to run CIME on Zeus. We need:

- CESM

  ```
  $ git clone -b cesm2.1-nemobranch https://github.com/ESCOMP/\
    cesm.git my_cesm2
  ```

- CIME and the other models
  - Edit Externals.cfg

    ```
    [cime]
    hash = 5dcb592
    protocol = git
    repo_url = https://github.com/ESMCI/cime
    local_path = cime
    required = True
    ```

  - Run:

    ```
    $ ./manage_externals/checkout_externals
    ```

NOTE: we are skipping CIME dependencies, since they are already satisfied on Zeus. Just remember to enable conda for python 2.7.

Table of contents

Introduction

CIME

Setting up the environment

Workflow

Homework

Workflow

The main steps to create and run a coupled model with CIME are:

1. Creation of a case
2. Setting up the case
3. Building the case
4. Submitting the case

In the following slides, we will analyse in detail each of these steps.

Creation of a case

Create the new case:

```
$ ./scripts/create_newcase --compset C_NEMO --res T62_n13 \
  --case HELLOWORLD --mach zeus --run-unsupported
```

The mandatory parameters are:

- **compset**: a component set defining all the involved models. To have a list of the available compsets:

  ```
  $ ./scripts/query_config --compsets
  ```

Creation of a case

Create the new case:

```
$ ./scripts/create_newcase --compset C_NEMO --res T62_n13 \
  --case HELLOWORLD --mach zeus --run-unsupported
```

The mandatory parameters are:

- **compset**: a component set defining all the involved models. To have a list of the available compsets:

  ```
  $ ./scripts/query_config --compsets
  ```

- **case name**: a friendly name for the case.

Creation of a case

Create the new case:

```
$ ./scripts/create_newcase --compset C_NEMO --res T62_n13 \
  --case HELLOWORLD --mach zeus --run-unsupported
```

The mandatory parameters are:

- **compset**: a component set defining all the involved models. To have a list of the available compsets:

  ```
  $ ./scripts/query_config --compsets
  ```

- **case name**: a friendly name for the case.
- **grid**: a string identifying a combination of the grids used by the models. A list of the available grids can be retrieved with:

  ```
  $ ./scripts/query_config --grids
  ```

Details. . .

- Compset C_NEMO is defined as:
  - `2000_DATM%NYF_SLND_DICE%SSMI_NEMO_DROF%NYF_SGLC_SWAV`
  - . . . that means NEMO as a prognostic model, atmosphere, ice and river as data models;
  - The rest is defined as a set of stub models.

Details. . .

- Compset C_NEMO is defined as:
  - `2000_DATM%NYF_SLND_DICE%SSMI_NEMO_DROF%NYF_SGLC_SWAV`
  - . . . that means NEMO as a prognostic model, atmosphere, ice and river as data models;
  - The rest is defined as a set of stub models.
- The grid is defined as:
  - T62 for atmosphere and land, tn1v3 for ocean and ice

Setting up the case

To setup the case, let's move to the case directory and:

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

This scripts creates some additional directories, configuration files and scripts.

Building the case

Now it's time to build the model, so (still from the case root):

```
$ ./case.build
```

If the process runs smoothly, at the end of the process we will have a single executable called `cesm.exe`.

Running the case

Once the model has been compiled, we can run it by submitting the request to the scheduler:

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

This process runs the coupled model and the following task called `archive`. We can check the status of the processes through `bjobs`.

Optional steps

- Modifying variables:

```
$ ./xmlchange VARIABLE VALUE
```

Optional steps

- Modifying variables:

  `$ ./xmlchange VARIABLE VALUE`

- Modifying the namelists: we edit the namelists called `user_nl_`xx where xx is the model

Optional steps

- Modifying variables:

  ```
  $ ./xmlchange VARIABLE VALUE
  ```

- Modifying the namelists: we edit the namelists called `user_nl_`xx where xx is the model
- Previewing the namelists:

  ```
  $ ./preview_namelists
  ```

Optional steps

- Modifying variables:

  `$ ./xmlchange VARIABLE VALUE`

- Modifying the namelists: we edit the namelists called `user_nl_`xx where xx is the model
- Previewing the namelists:

  `$ ./preview_namelists`

- Previewing the run:

  `$ ./preview_run`

Optional steps

- Modifying variables:

  `$ ./xmlchange VARIABLE VALUE`

- Modifying the namelists: we edit the namelists called `user_nl_`xx where xx is the model
- Previewing the namelists:

  `$ ./preview_namelists`

- Previewing the run:

  `$ ./preview_run`

- Getting the list of components:

  `$ ./query_config --components`

Table of contents

Introduction

CIME

Setting up the environment

Workflow

Homework

Homework

For the next week, I suggest to:

- Study something about CIME
  - *CIME Documentation*
- Study something about WRF
  - *An introduction to WRF Modeling System*
- Study something about NEMO
  - *COST-EOS training: Ocean Modeling the Nemo model at high resolution*

NOTE: An in-depth study is not required, you just to understand what these models/tools are and what's their purpose.