CERFACS

EUROPEAN CENTRE FOR RESEARCH AND ADVANCED TRAINING IN SCIENTIFIC COMPUTING

Code coupling for climate modelling

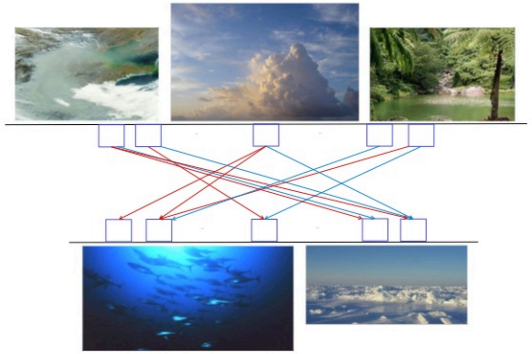
S. Valcke

August 5 2016

Outline

- How to run a coupled system on a computing platform?
- Two main technical approaches to coupling
- 1- Coupling framework integrated approach
  - ESMF
  - CESM/cpl7
  - FMS
- 2- Coupler or coupling library approach
  - OASIS
- Conclusions

Code coupling in climate modelling



Why couple ocean and atmosphere (and sea-ice and land and …) models?

- Of course, to treat the Earth System globally

What does "coupling of codes" imply?

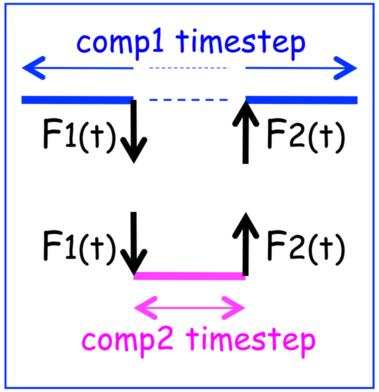
- Exchange and transform information at the code interface
- Manage the execution and synchronization of the codes

What are the constraints?

- ✓ **Physical constraints**: e.g. energy conservation at the interfaces
- ✓ **Coupling algorithm** dictated by **science** and **modeling**
- ✓ Coupling should be **easy to implement**, **flexible**, **efficient**, **portable**
- ✓ Start from **existing** and **independently** developed codes
- ✓ Global **performance** and **load balancing** issues are crucial

How to run a coupled system on a computing platform?

Sequential coupling :

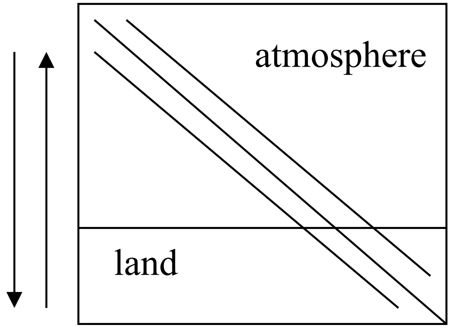


Implicit resolution of heat diffusion equation from the top of the atmosphere to the bottom of the land

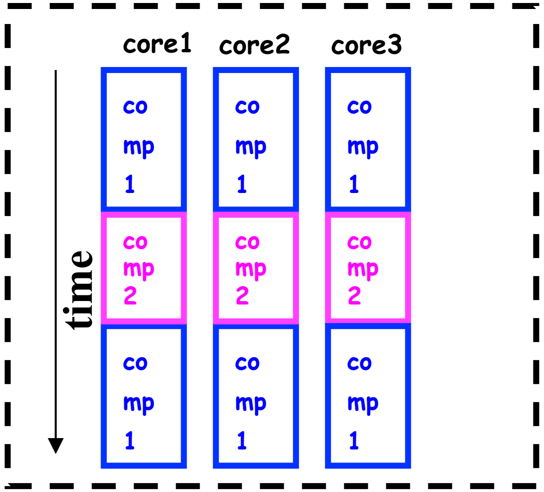
$$\partial T / \partial t = K \, \partial^2 T / \partial z^2$$

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} + 2T_k^{n+1}}{\Delta z^2}$$

$$AT^{n+1} = T^n$$



=> sequential execution on the same set of cores in one executable



☺ Efficient coupling exchanges through the memory

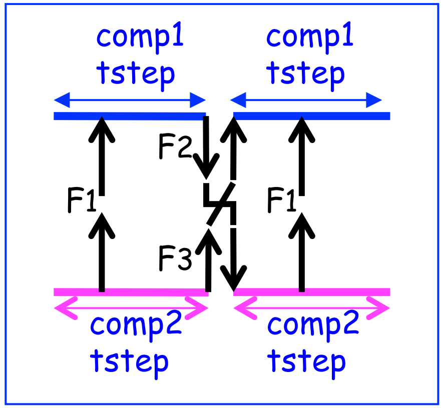
☺ Optimal for load balancing if components can run efficiently on same number of cores

☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)

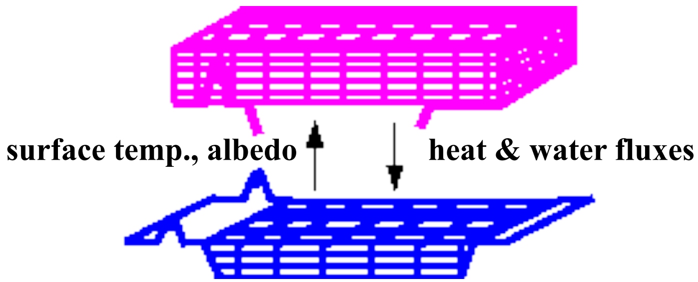
☹ No flexibility in coupling algorithm

How to run a coupled system on a computing platform?

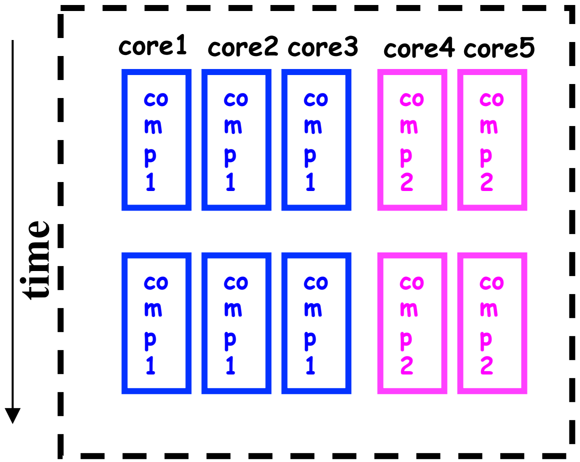
Concurrent coupling:



Traditional asynchronous ocean-atmosphere coupling



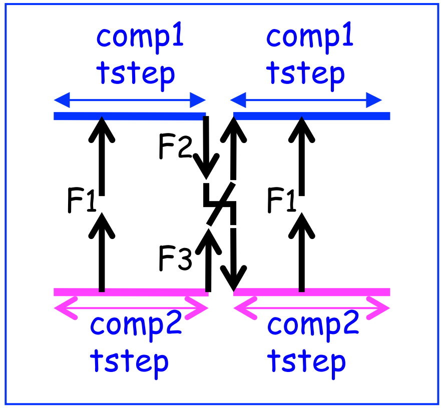
=> concurrent execution on different sets of cores within one executable



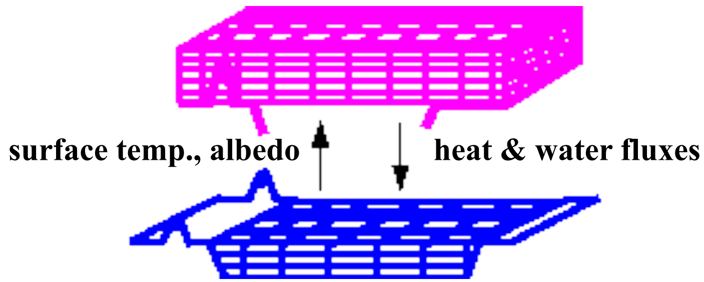
- ☺ Flexible coupling algorithm (exchanges in timestep)
- ☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)
- ☹ Harder load balancing

How to run a coupled system on a computing platform?

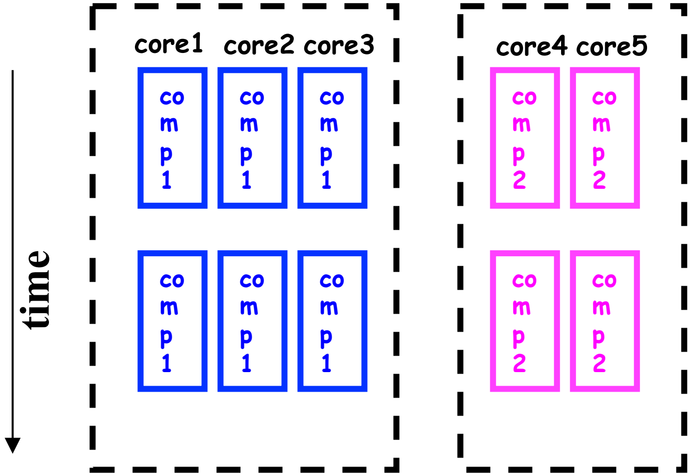
Concurrent coupling:



Traditional asynchronous ocean-atmosphere coupling



=> concurrent execution on different sets of cores within separate executables



- ☺ Flexible coupling algorithm (exchanges in timestep)
- ☺ No conflicts as components remain separate executables (I/O, units, internal comm, etc.)
- ☺ Optimal use of memory
- ☹ Harder load balancing
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)

Outline

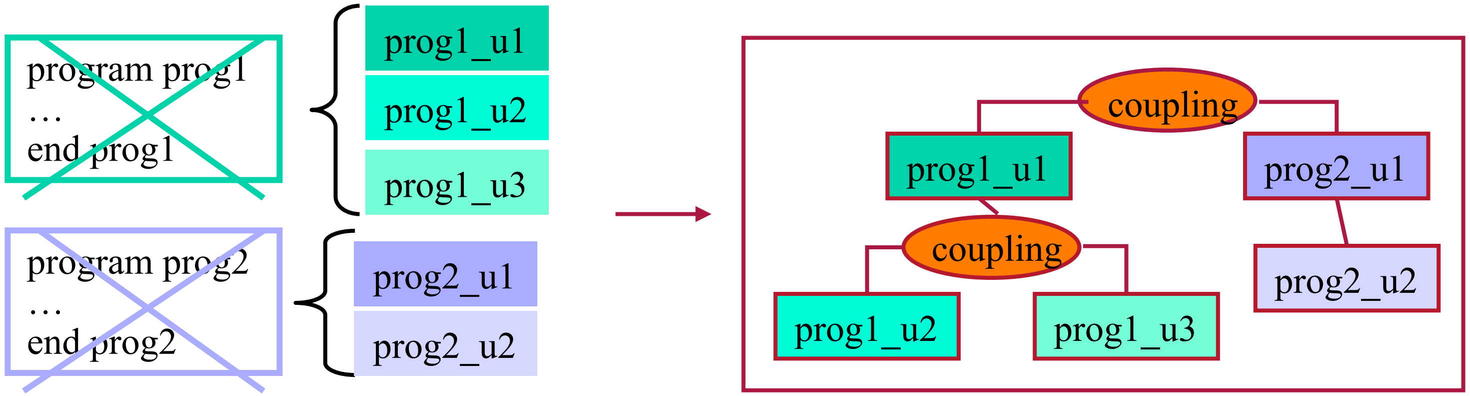
- How to run a coupled system on a computing platform?
- **Two main technical approaches to coupling**
- 1- Coupling framework integrated approach
    - ESMF
    - CESM/cpl7
    - FMS
- 2- Coupler or coupling library approach
    - OASIS
- Conclusions

Two main technical approaches to coupling

1. Coupling framework integrated approach

- Split code into elemental units
  at least init/run/finalize
- Write or use coupling units

- Adapt data structure and calling interface
- Use the framework (to build) a hierarchical merged code



☺ efficient
☺ sequential and concurrent components
☺ use of generic utilities (parallelisation, regridding, time management, etc.)

☹ existing codes
😐 (easy)

ESMF

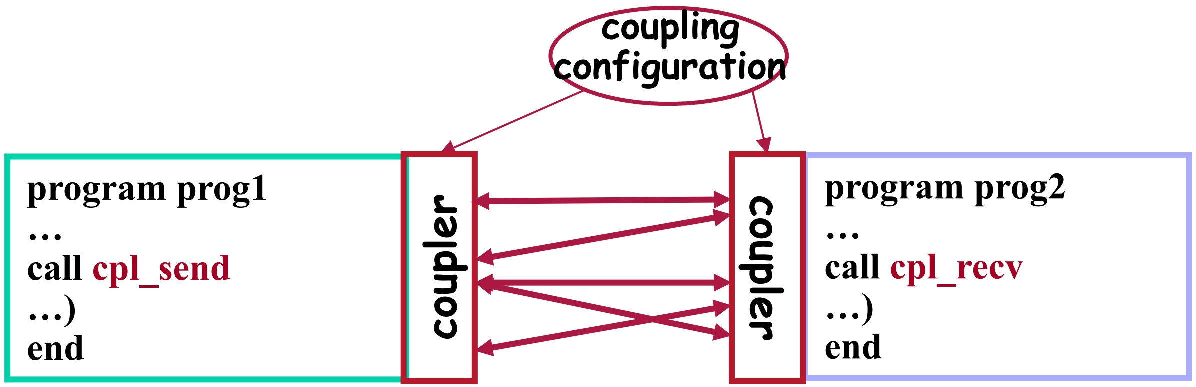
FMS (GFDL)

CESM (NCAR)

→ probably best solution in a controlled development environment

Two main technical approaches to coupling

2. Coupler or coupling library approach



☺ existing codes
☺ use of generic transformations/ regridding
☺ concurrent coupling (parallelism)

☹ Sequential components: waste of resources?
☹ multi-executable: more difficult to debug; harder to manage for the OS
😐 efficient

Oasis O-Palm mct

→ probably best solution to couple independently developed codes

Outline

- How to run a coupled system on a computing platform?
- Two main technical approaches to coupling
- 1- Coupling framework integrated approach
  - ESMF
  - CESM/cpl7
  - FMS
- 2- Coupler or coupling library approach
  - OASIS
- Conclusions

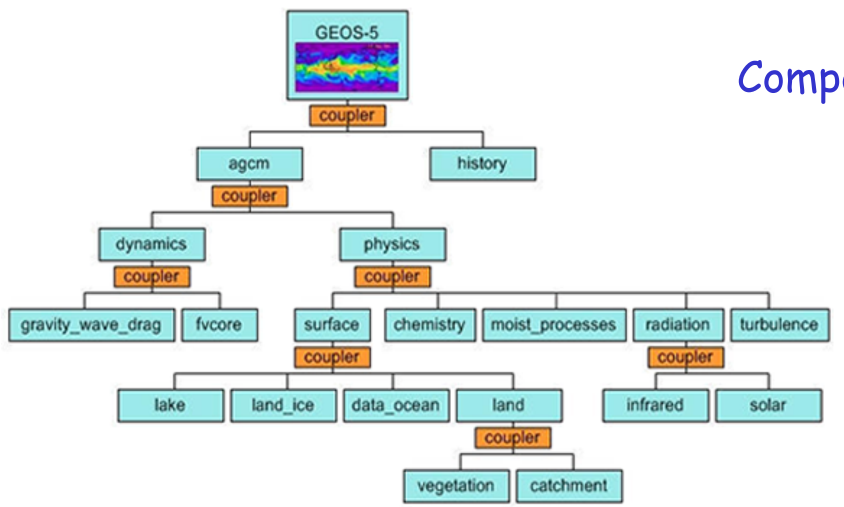
Coupling framework integrated approach - ESMF

Earth System Modeling Framework

Open source software for building climate and weather applications based on components developed in different modeling centers

- Multi-agency governance (NSF, NASA, DoD, NOAA) with many partners
- Mainly written in C++, with F90 and Python interfaces
- Run nightly on 24+ platforms using a suite of over 4000 tests
- 13 different modelling systems, ~ 80 different components
- NUOPC layer: US NWP centers conventions and templates for better interoperability

Component-based design:



Goddard Space Flight Center GEOS-5 model

Component = well-defined calling interface + coherent function

- Gridded Components: scientific code
- Coupler Components: data transformation/transfer

➢ user builds a model as hierarchy of components
➢ can be run sequentially, concurrently, in mixed mode
➢ single executable

Coupling framework integrated approach - ESMF

Hill et al., Comput. Sci. Eng., 2004

ESMF "Infrastructure" :

- calendar management; message logging, data regridding & communication
- regridding weight generation: bilinear, patch, or first order conservative methods up to 3D (can be used off-line)

ESMF "Superstructure": coupling tools and component wrappers with standard interfaces

1. Define Gridded Components : slip code into init, run and finalize methods
2. Wrap native data structures into ESMF data structure
3. Write Coupler Components
4. Register init, run and finalize methods to ESMF comp (in driver application)
5. Schedule components and exchange data
6. Execute the application

```
subroutine myOceanRun (.. , impState, expState, clock, …)
 type(ESMF_State)    :: impState
```

```
subroutine  oceanToAtmCpl (.. ,)
 call ESMF_FieldRedist(oceanField, atmField, …)
```

```
…
call ESMF_GridCompSetEntryPoint (oceanComp,
     ESMF_SETRUN, myOceanRun, …)
…
call ESMF_GridCompRun(oceanComp,  …)
call ESMF_CplCompRun (oceanToAtmCpl, …)
call ESMF_GridCompRun(atmComp,  …)
```

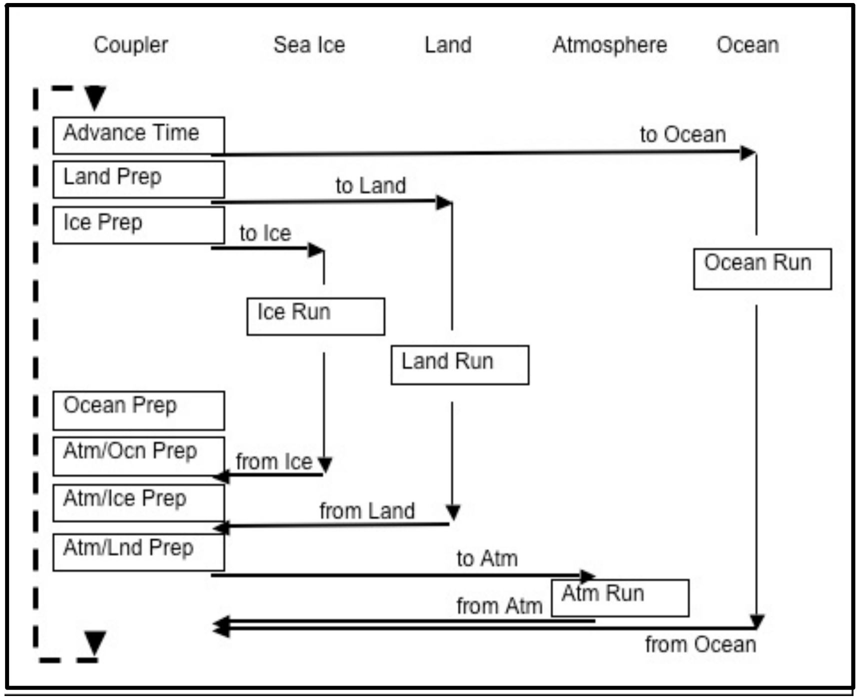
Coupling framework integrated approach – CPL7

Cpl7 for CCSM4 and CESM1

Software top-level layer (driver), that calls a coupler component and atmosphere, ocean, land and sea ice codes in sequence or in parallel

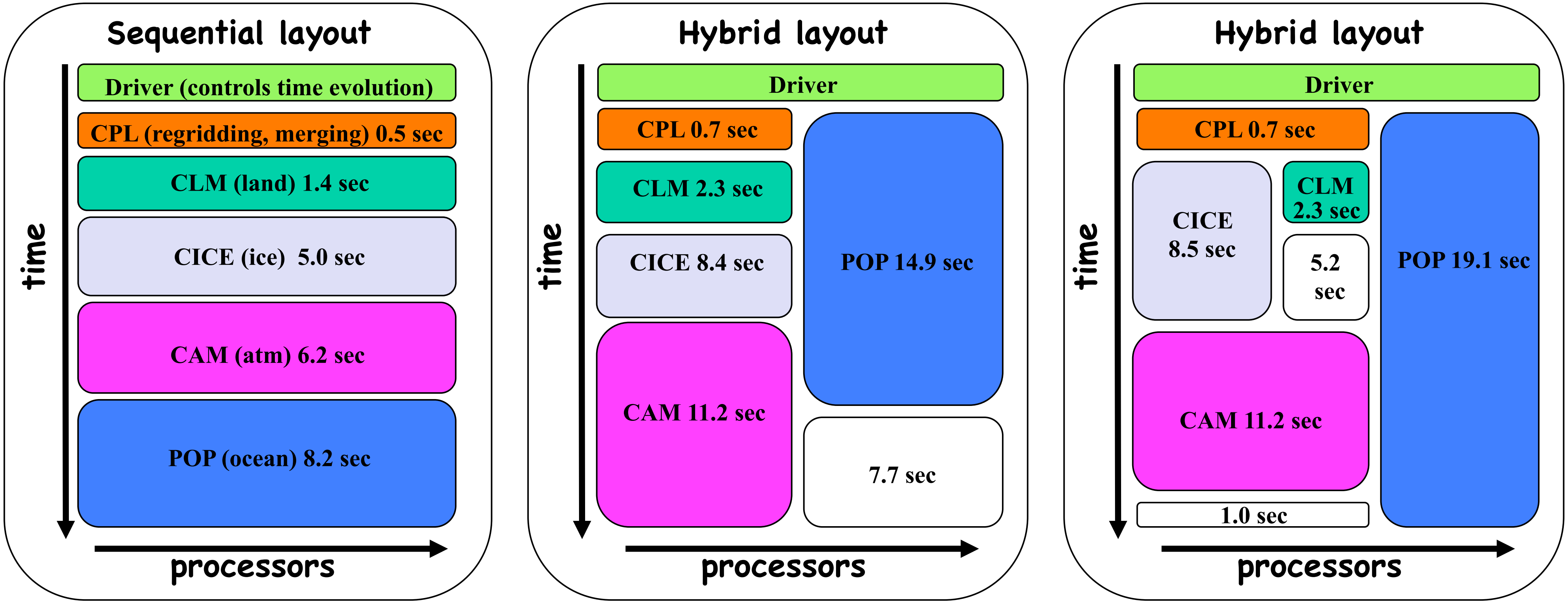
- Developed by the NCAR Earth System Laboratory,
- From multiple concurrent executables (cpl6) to one executable: time flow easier to understand, easier to debug
- Ability to add new components, new coupling fields
- Interface compatibility for ESMF-compliant components
- Ported to IBM p6, Cray XT4/XT5, BGP, Linux Clusters, SGI

Driver Loop Sequencing



Coupling framework integrated approach – CPL7

➢ Varying levels of parallelism via external configuration (metadata) for proc layout:



➢ Scaling evaluated on up to 10 000 processors:

Craig et al., Int. J. High Perform. C, 2012

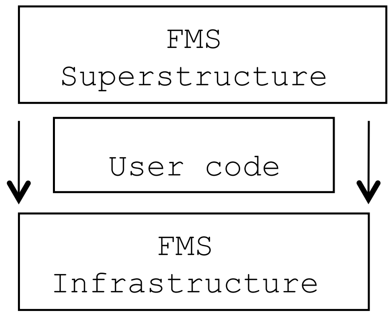
- flop intensive kernels: linear
- memory intensive operations: linear at low proc counts, flattens at high proc counts
- comm-dominated kernels: sub-linear at low proc counts; drops off for + 1000 procs.

Coupling framework integrated approach – FMS

The Flexible Modeling System (FMS)

Software to assemble a climate model with domain-specific "slots" for atmosphere, ocean, ocean surface including sea ice and land surface

- Active for over a decade at GFDL; developed in F90
- FMS shown to be scalable with up to O(10000) pes



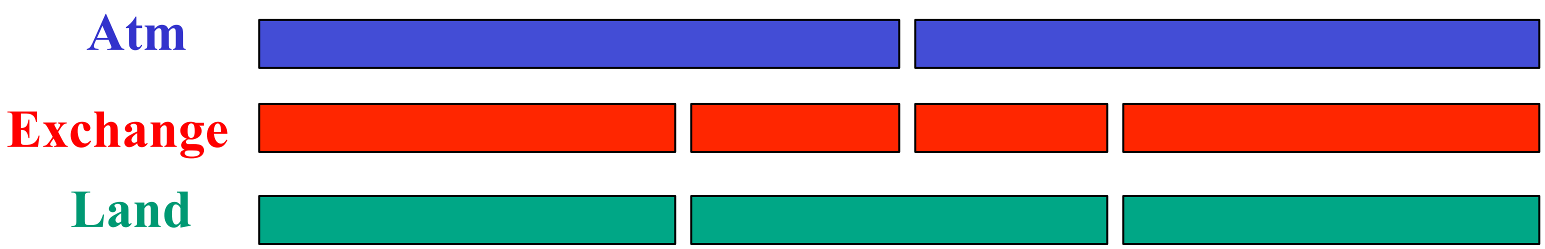
- FMS "Infrastructure": I/O, except. handling, operations on distributed gridded fields (expressed independently of the underlying platform)

- FMS "Superstructure":
  - Domain-specific coupling layer ("stubs" (no component), or "data" also possible)
  - Components "wrapped" in FMS-specific data structures and procedure calls
  - Single executable with serial or concurrent execution of components
  - Regridding, redistribution, or direct (hard-coded) exchanges between components

Coupling framework integrated approach – FMS

FMS "Superstructure" obeys specific geophysical constraints

- Interface fluxes must be globally conserved
  - atmosphere water-land fractions adjusted to fit ocean sea-land mask
  - quantities are transferred from the parent grids to the *exchange grid*, where fluxes are computed; they are then averaged on the receiving grid

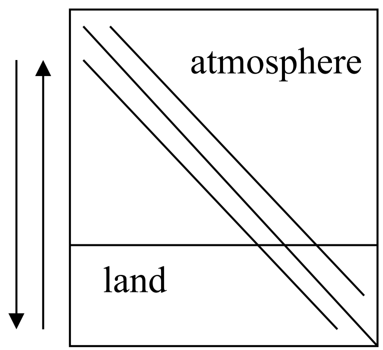


- Exchanges consistent with physical processes occurring near the surface
  - Implicit calculation of vertical diffusive fluxes over the whole column
  - Up-down sweep for tridiagonal matrix resolution through the exchange grid

$$\partial T / \partial t = K \, \partial^2 T / \partial z^2$$

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} + 2T_k^{n+1}}{\Delta z^2}$$

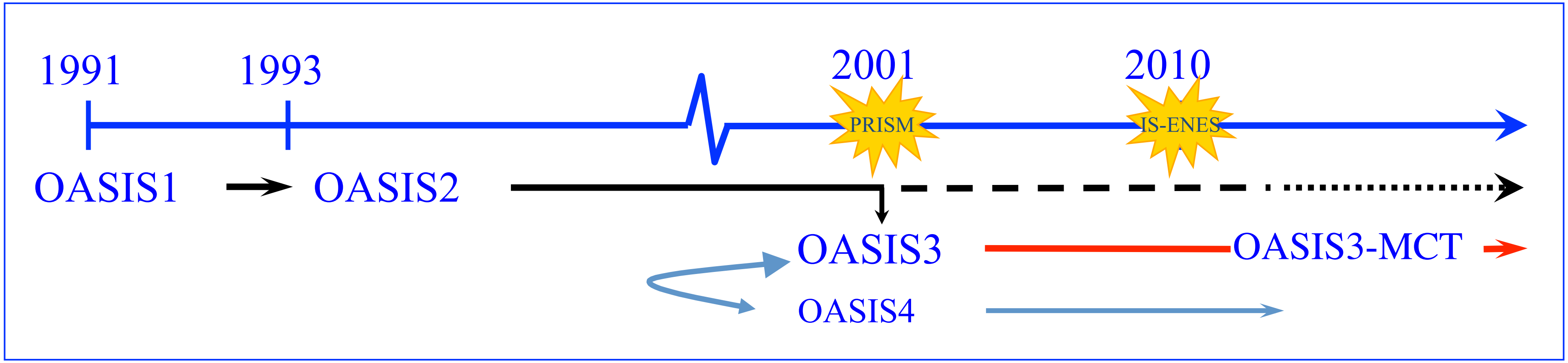
$$A T^{n+1} = T^n$$



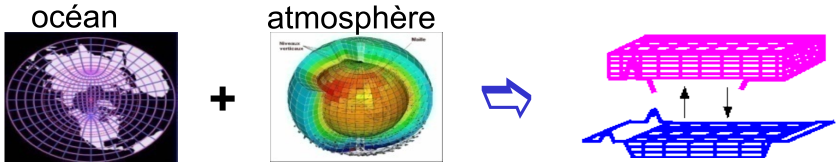
Outline

- How to run a coupled system on a computing platform?
- Two main technical approaches to coupling
- 1- Coupling framework integrated approach
  - ESMF
  - CESM/cpl7
  - FMS
- 2- Coupler or coupling library approach
  - OASIS
- Conclusions

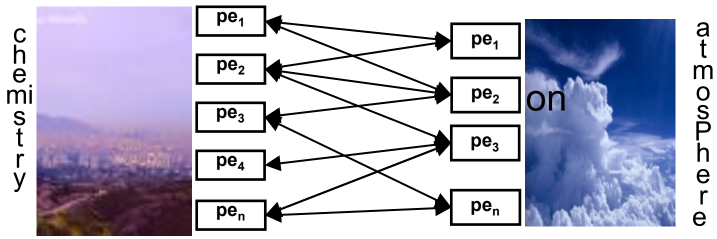
2- Coupler or coupling library approach - OASIS



- **OASIS1 -> OASIS2 -> OASIS3:**
  2D ocean-atmosphere coupling
  low resolution, low frequency
  → flexibility, modularity, 2D interpolations



- OASIS4 / **OASIS3-MCT:**
  2D/3D coupling of high resolution parallel components
  massively parallel platforms
  →parallelism, efficiency, performance



2- Coupler or coupling library approach – OASIS

OASIS community today: more than 40 groups in France, Europe and around the world for climate modelling and seasonal prediction, e.g.:

- France: CERFACS,CNRM, LOCEAN, LMD, LSCE, LA, LATMOS, LEGOS, LGGE, IFREMER, ENSTA
- Europe: ECMWF + EC-Earth community
- Germany: MPI-M, IFM-GEOMAR, HZG, U. Frankfurt, BTU-Cottbus
- UK: MetOffice, NCAS/U. Reading, ICL,
- Denmark: DMI
- Norway: U. Bergen
- Sweden: SMHI, U. Lund
- Ireland: ICHEC, NUI Galway
- Netherlands: KNMI
- Belgium: KU Leuven
- Switzerland: ETH Zurich
- Italy: INGV, ENEA, CASPUR
- Czech Republic :CHMI
- Spain: IC3, BSC, U. Castilla
- Tunisia: Inst. Nat. Met

- Saudi Arabia: CECCR
- Japan: U. Tokyo, JMA, JAMSTEC
- China: IAP-CAS, Met. Nat. Centre, SCSIO
- Korea: KMA
- Australia: CSIRO, BoM, ACT, NCI
- New-Zealand: NIWA, NCWAR
- Canada: Environment Canada, UQAM
- USA: Oregon St. U., Hawaii U., JPL, MIT
- Peru: IGP

+ downloads from Nigeria, Colombia, Singapore, Russia, Thailand, Chili,Iran, …

2- Coupler or coupling library approach – OASIS

Communication and regridding library to exchange data between independent models with minimal level of interference in the codes (external configuration through namelist-like file)

- Developed by CERFACS since 1991 with CNRS since 2005 and many others
- Written in F90 and C; open source license (LGPL)
- Last OASIS3-MCT version based on MCT
- Public domain libraries: MPI; NetCDF; LANL SCRIP

Application Prog Interface

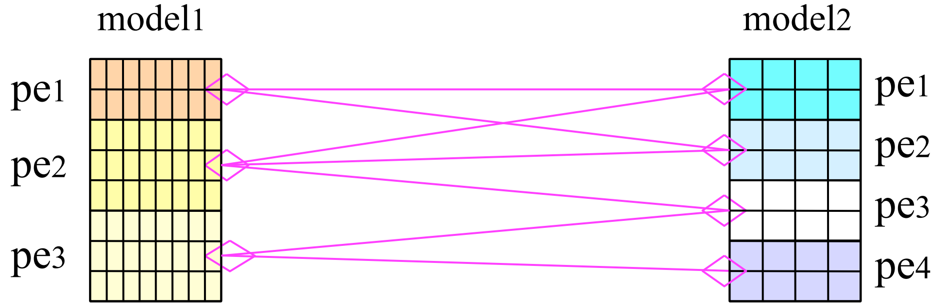
- Initialization: `call oasis_init(...)`
- Grid definition: `call oasis_write_grid (...)`
- Local partition definition: `call oasis_def_partition (...)`
- Coupling field exchange: in model time stepping loop

```
call oasis_put (…, time, var_array. …)
call oasis_get (…, time, var_array, …)
```

- user external configuration:
  - => source or target (end-point communication)
  - => effective coupling frequency
  - => transformations and regridding

OASIS3-MCT communication

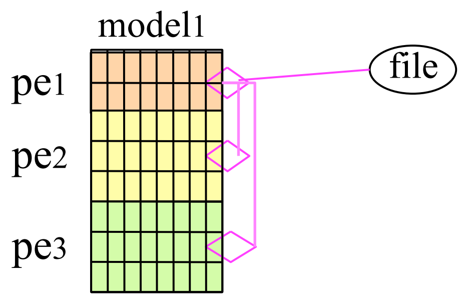
- Fully parallel communication between parallel models based on Message Passing Interface (MPI)



If required, the interpolation weights and addresses are calculated onto one model process

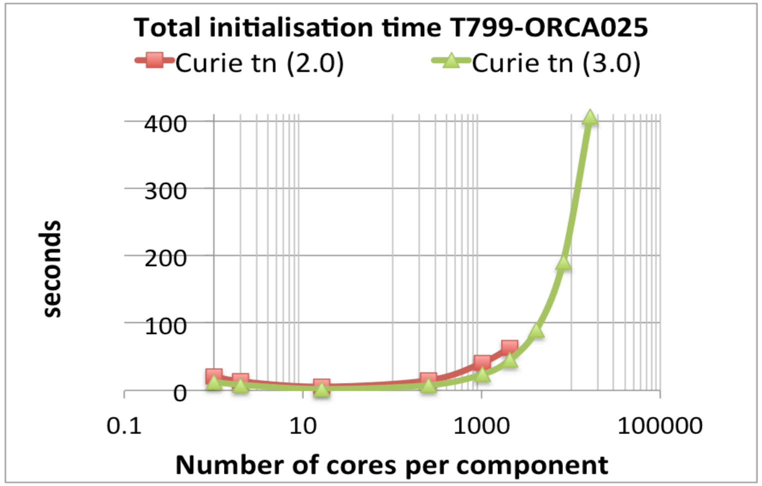
Interpolation per se from the source grid to the target grid is done in parallel on the source or on the target processes

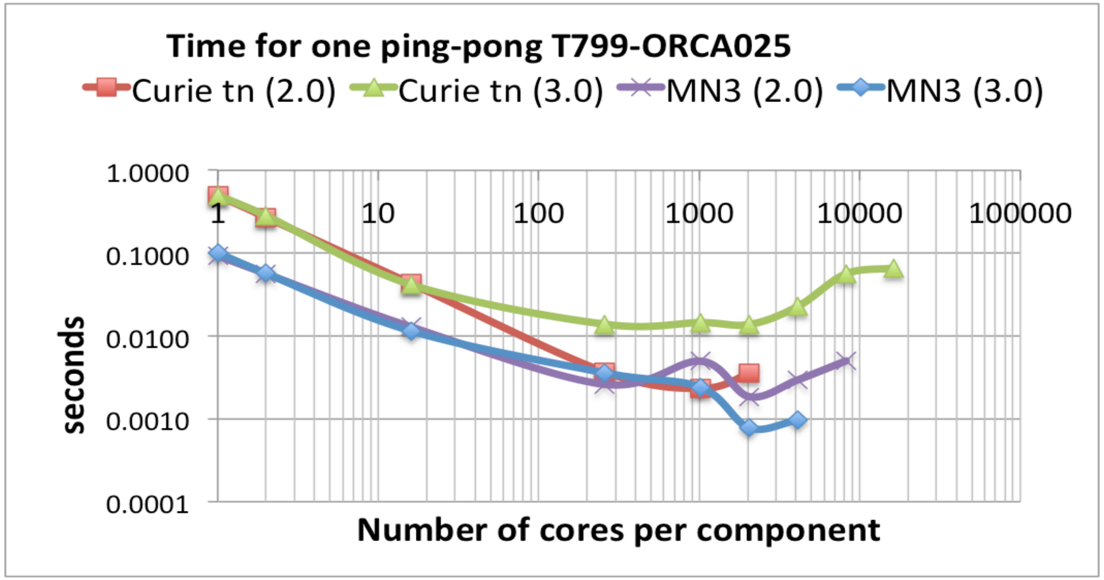
- I/O functionality (switch between coupled and forced mode):



OASIS3-MCT performance

- Toy coupled model: NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes; IBM MareNostrum3





Coupling overhead for one-year long simulation with one 1 coupling exchange every 3 hours in each direction between codes with O(1 M) grid points running on O(10K) cores/component:

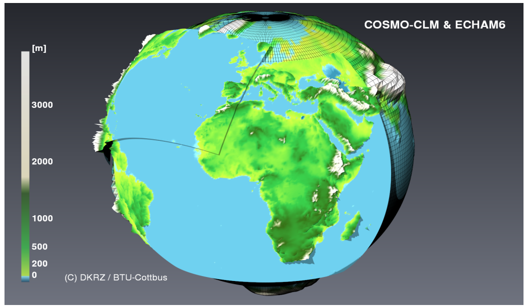
- ~7 mins for initialisation, ~5 mins for data exchange

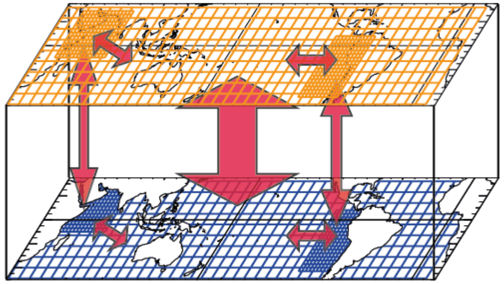
2- Coupler or coupling library approach - OASIS

Some heavy coupled applications using OASIS

3D global-regional atmosphere coupling

- BTU-Cottbus (Germany)
- **COSMO-CLM -regional** (221x111x47, ~2 deg) - **ECHAM -global** (T63, 192x96x47), + 2D coupling with MPI-OM ocean (254x220)
- 6 x 3D coupling fields every global atm timestep



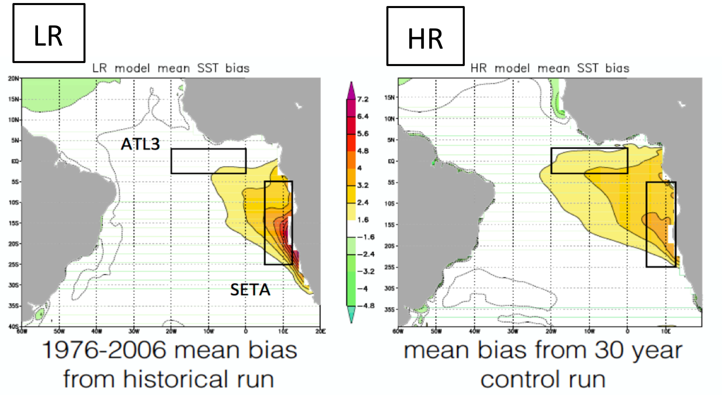


HR tropical coupled modeling

- **WRF** - **NEMO**, zooms in both models, 27-9 km (4322x1248 pts),
- ANR & PRACE PULSATION project 22 Mhrs Bullx Curie.

Seasonal & decadal climate forecasting with CERFACS-HR

- **NEMO** (ocean ¼ degree) - **ARPEGE** (atmosphere ~50 kms)
- SPRUCE : PRACE project , 27 Mhrs, Bullx Curie TGCC
- HiResClim: PRACE project , EU project SPECS, 10 Mhrs, IBM MareNostrum3 BSC
- Follow-on: CNRM-CM6_HR for CMIP6 and PRIMAVERA EU project



The coupled system: high-resolution model CERFACS-HR SPECS

SPECS
Seasonal-to-decadal climate Prediction for the improvement of European Climate Services

PRACE

Arctic sea-ice extent, 1994-2003

Outline

- How to run a coupled system on a computing platform?
- Two main technical approaches to coupling
- 1- Coupling framework integrated approach
  - ESMF
  - CESM/cpl7
  - FMS
- 2- Coupler or coupling library approach
  - OASIS
- Conclusions

Conclusions

- Technical choices affect the load balancing and performance of the system:
  - One executable vs many executable
  - Sequential vs concurrent execution of the components
  - Same or different set of cores (number of cores per component?)

- Different coupling approaches are used in climate modelling:
  - Integrated approach: split original code into init/run/finalize
    - use a "standard" methods to build coupled system (e.g. ESMF)
    - integrate in a predefined driving layer (e.g. FMS, Cpl7)

    ➢ **more efficient** in many cases but puts **more constraints** on the components
  - External coupler and/or communication library (e.g. OASIS):

    ➢ **easiest** solution to couple independent codes but some performance **drawbacks**

- The "best" coupling tool is not uniquely defined; it depends on:
  - readiness to change/adapt your original codes
  - efficiency you want/need to achieve, etc.
  - coding environment and interactions!

Implementing (climate) coupled models is not an easy task !

The end