A review of the coupling technologies used in climate modelling

S. Valcke - CERFACS

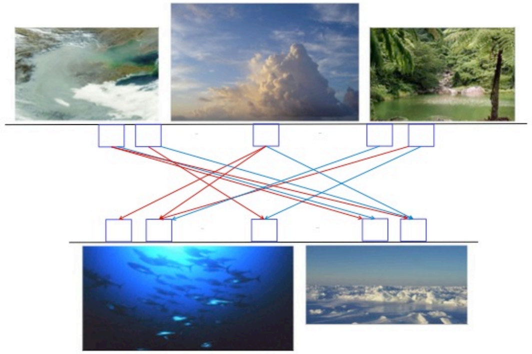
S. Valcke, V. Balaji, A. Craig, C. Deluca, R. Dunlap, R. Ford, R. Jacob, J. Larson, R. O'Kuinghttons, G. Riley, M. Vertenstein, 2012: Coupling technologies for Earth System Modelling, GeoscMD., 5, 1589-1596, doi:10.5194/gmdd-5-1589-2012.

S. Valcke, R. Redler, C. DeLuca, G. Theurich, V. Balaji, C. Linstead, R. Jacob, J. Larson, R. Ford, G. Riley. Brief on "Earth System Modelling – Volume 4 - Coupling software and strategies", R. Budich and R. Redler Eds., Springer, December 2011.

Outline

- Code coupling in climate modelling
- Two main approaches to coupling
- Some coupling technologies used in climate modelling
    - ESMF
    - CESM/cpl7
    - FMS
    - OASIS
    - MCT, TDT, BFG, Open-Palm
- 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?

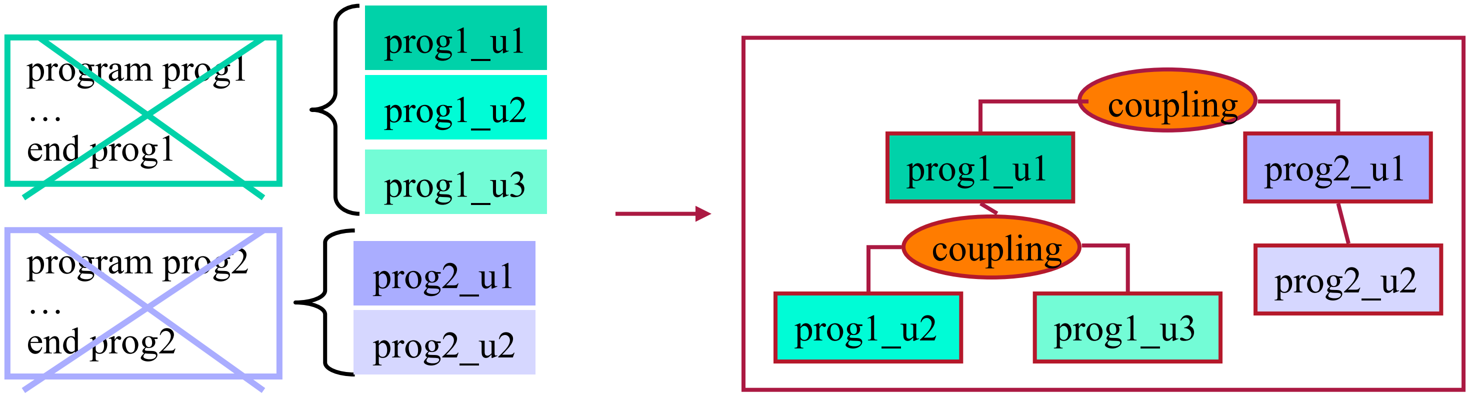
- ✓ **Physical constraints**: e.g. energy conservation at the interfaces
- ✓ **Coupling algorithm** dictated by **science**
- ✓ Coupling should be **easy to implement**, **flexible**, **efficient**, **portable**
- ✓ Start from **existing** and **independently** developed codes
- ✓ Global **performance**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

Two main 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



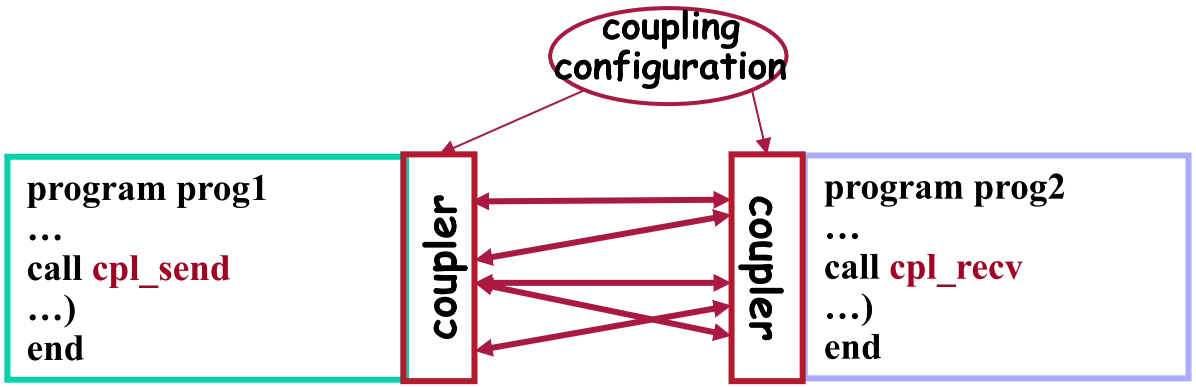
☺ efficient
☺ sequential and concurrent components
☺ use of generic utilities (parallelisation, regridding, time management, etc.)

☹ existing codes
😐 (easy)

Ex: with built-in driving layer: CESM, FMS
without pre-defined driving layer: ESMF

Two main approaches to coupling

2. Coupler or coupling library approach



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

- ☹ multi-executable: possible waste of resources if sequential execution of the components is enforced
- ☹ multi-executable: more difficult to debug; harder to manage for the OS
- 😐 efficient

Ex: OASIS, TDT, Open-Palm

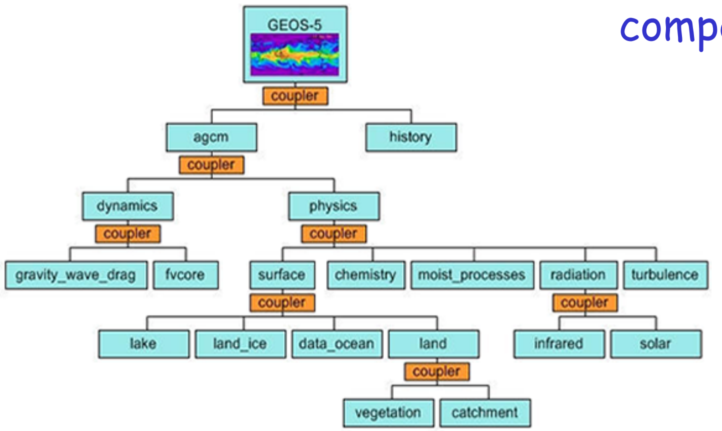
Current coupling technologies - ESMF

Earth System Modeling Framework

ESMF

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
- 12 different modelling systems, totaling about 85 different components
- NUOPC layer: US NWP centers conventions and templates (model, simple connector, mediator, driver) for better interoperability – release early 2013
- 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 nested components
  - can be run sequentially, concurrently, in mixed mode
  - single executable
  - support for multiple executables and web services

Current coupling technologies - ESMF

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,  …)
```

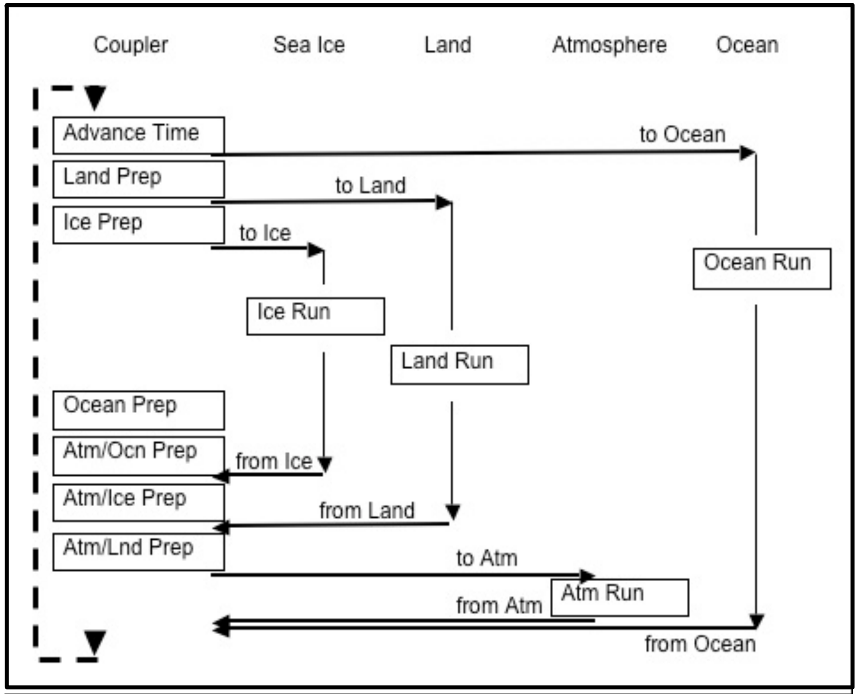
Current coupling technologies – Cpl7

Cpl7 for CCSM4 and CESM1

Software architecture with top-level driver and coupler component for flexible assembling of atmosphere, ocean, land and sea ice models into one executable via standard init/run/finalise interfaces

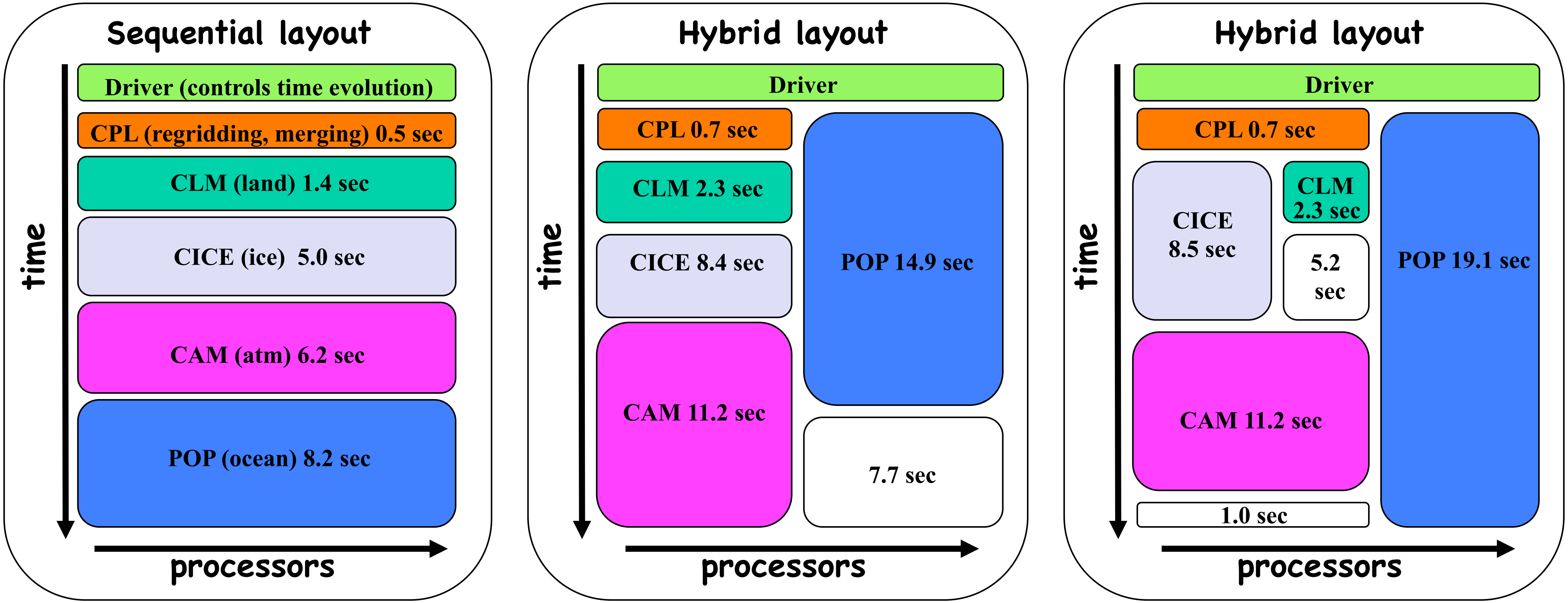
- Developed by the NCAR Earth System Laboratory, uses Argonne Nat Lab MCT for data regridding and exchange
- From multiple concurrent executables (cpl6) to one executable: time flow easier to understand, easier to debug
- Ability to add new components, new coupling fields, new capabilities (e.g. data assimilation) ; interface compatibility for ESMF-compliant components
- Ported to IBM p6, Cray XT4/XT5, BGP, Linux Clusters, SGI

Driver Loop Sequencing



Current coupling technologies - Cpl7

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



- Scaling evaluated on up to 10 000 processors:

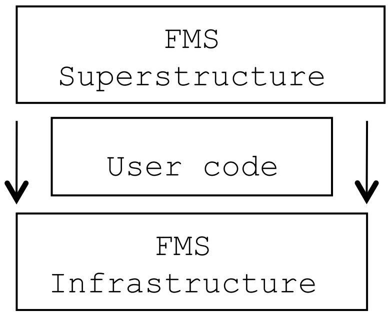
Craig et al., Int. J. High Perform. C, 2012

- 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.

Current coupling technologies – 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

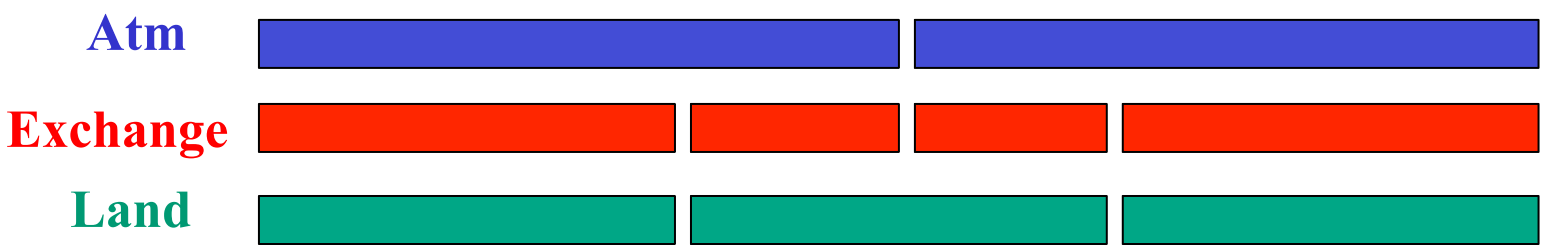


- Active for over a decade at GFDL; developed in F90
- Use in CMIP5: CM3 (interactive aerosol chem), ESM2M & ESM2G (carbon cycle), HiRAM-C180 & HiRAM C360 (high-res time-slices)
- FMS shown to be scalable with up to O(10000) pes
- FMS "Infrastructure": I/O, except. handling, operations on distributed 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
    - Includes data assimilation

Current coupling technologies - 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

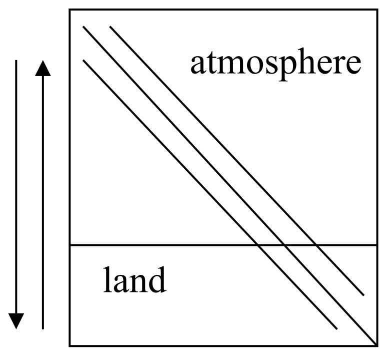


- Exchanges consistent with physical processes occurring near the surface
  - Implicit calculation of vertical diffusive fluxes over the whole column
  - Up-down sweep for tridiagonal matrix resolution through the exchange grid

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

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

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



Current coupling technologies - 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
- Large community of users: ~35 climate modelling groups world-wide

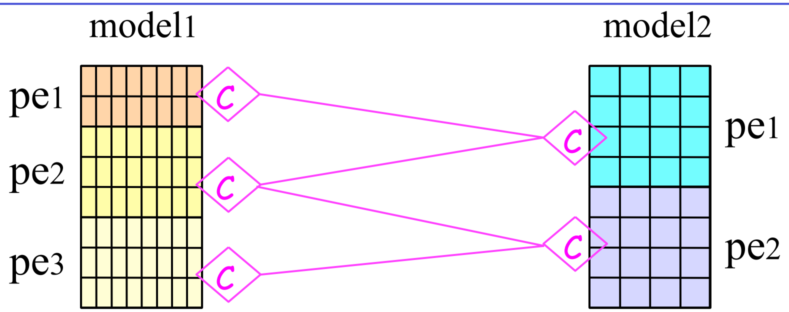
Application Prog Interface

- 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

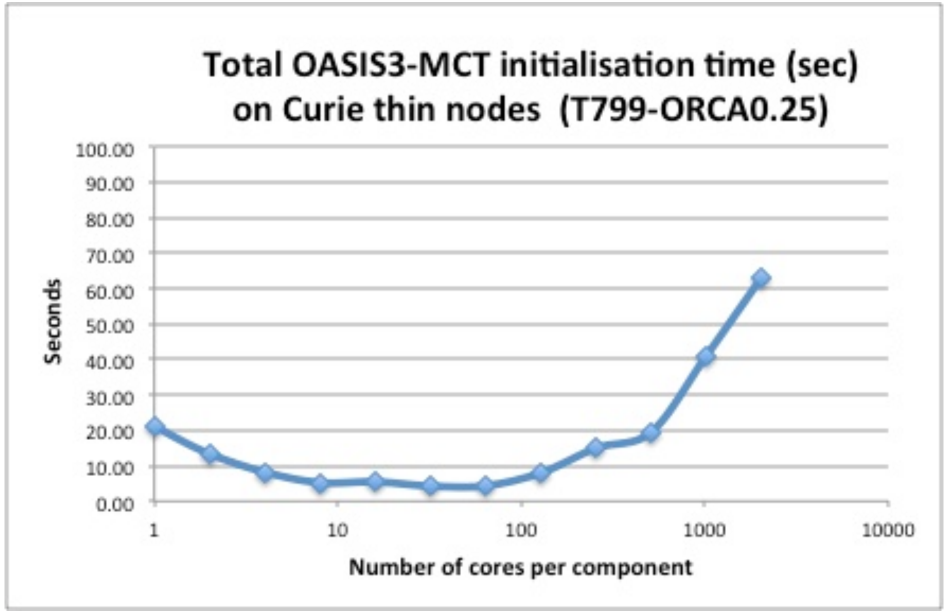
```
call oasis_put (…, time, var_array. …)
call oasis_get (…, time, var_array, …)
```

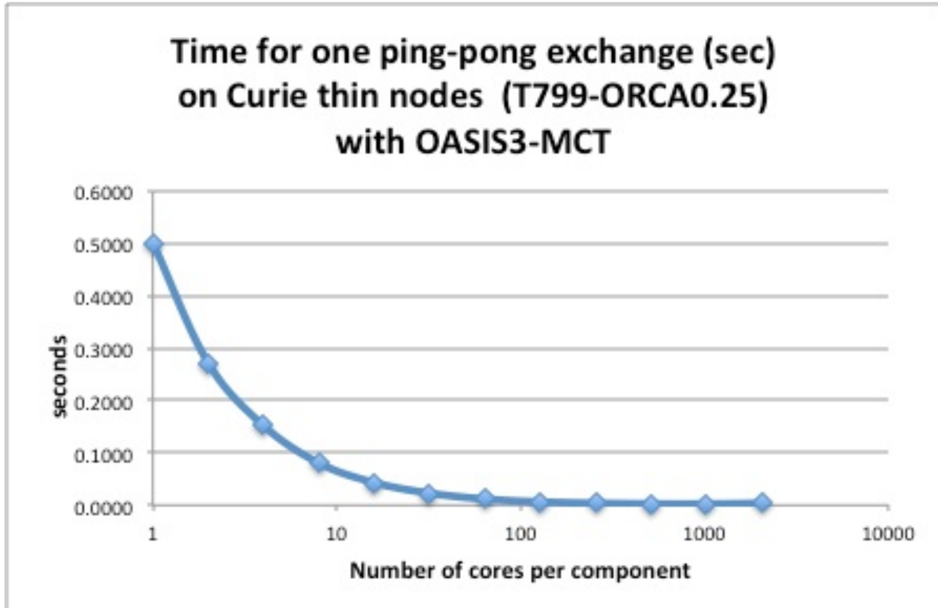
- user external configuration:
  - => source or target (end-point communication)
  - => effective coupling frequency
  - => transformations and regridding

Current coupling technologies - OASIS



- Sequential weights calculation (SCRIP library)
- Parallel regridding on source or targets processes (MCT)
- Redistribution of coupling data (MCT)





Current coupling technologies

Model Coupling Toolkit (Argonne National Lab):

- library and related datatypes for parallel field exchanges, parallel data transformation for creating parallel couplers / coupled models
- used in CESM/Cpl7 and OASIS3-MCT

www.mcs.anl.gov/research/projects/mct/

Typed Data Transfert (Potsdam Institute for Climate – PIK)

- library to transfer data between programs in a platform and language independent way with different protocols (Unix Socket, I/O from/to files, MPI) -> heterogeneous coupling

Bespoke Framework Generator (U. Manchester):

- Generator of wrapper code around components to build a coupled model with a chosen coupling technology (curr. OASIS, MPI, argument passing) based on external metadata for flexible model composition and deployment
- Used for 20 configurations in CIAS (Comm Integrated Assessment System)

www.cs.manchester.ac.uk/cnc/ projects/bfg

Open-PALM (CERFACS - ONERA)

- Tool originally designed for creating data assimilation suites
- Dynamic driver and CWIPI coupling library (ONERA) performing parallel data exchanges, parallel calculation of weights, parallel remapping
- Multiphysic coupling for ~40 different applications mainly in France

Conclusions

- Different coupling approaches used in climate modelling:
    - External coupler and/or communication library (e.g. OASIS):
        - **easiest** solution to couple independent codes but some **drawbacks**
    - 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
- The "best" coupling tool is not uniquely defined; it depends on:
    - ready to change/adapt your original codes
    - efficiency you want/need to achieve, etc.
- The code split into clean init/run/finalize part is recognized as a best practice and groups are encouraged to follow it!

The end