

OASIS3-MCT_3.0

*A. Craig, S. Valcke, L. Coquart, CERFACS
April 20-22, 2015*

- OASIS overview
- OASIS3-MCT_3.0
- Some recent performance results
- Summary and future efforts



What is OASIS?

- Coupling Software used by at least 35 projects, world-wide
- Based on an "inline" send/recv approach (get and put calls inside components)
- A few OASIS calls need to be added during component initialization and then fields are coupled via get and put calls during the run phase.
- Requires that components provide information to the OASIS coupling layer
 - grids
 - decompositions (partitions)
 - coupling variable names
- Coupling interactions defined in an input file, "namcouple"
 - associates variable names between components
 - defines mapping (regridding) operations
 - specifies lags and sequencing, coupling frequency
 - includes basic math functions (average, min, max, etc)
- Has I/O capability
 - OASIS can read forcing data and pass that data to components
 - OASIS can write coupling data for analysis and diagnosis

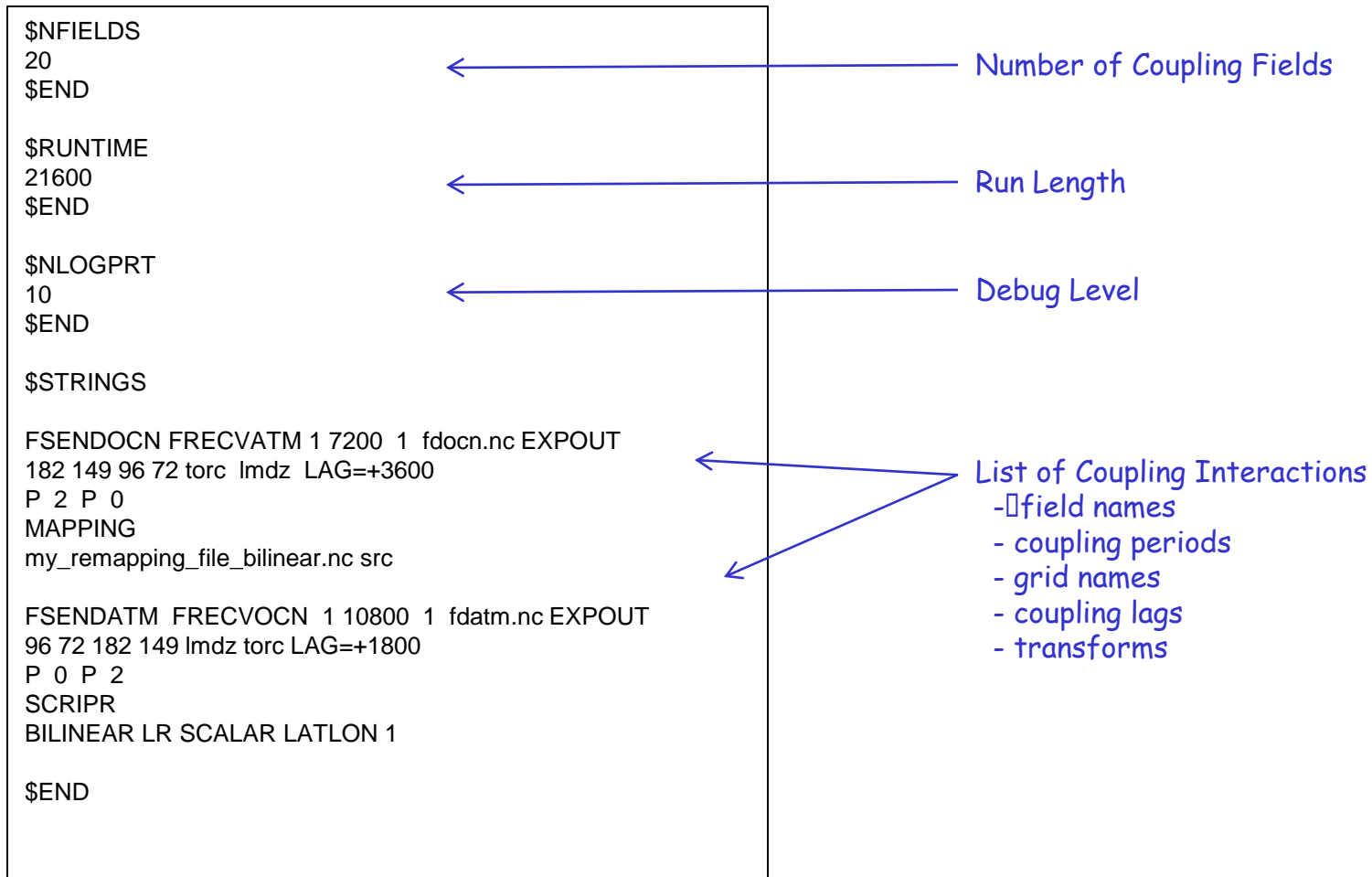


OASIS3-MCT: component interfaces

- Initialization: `call oasis_init_comp(...)`
- Grid definition: `call oasis_write_grid (...)`
- Local partition definition: `call oasis_def_partition (...)`
- Coupling field declaration: `call oasis_def_var (...)`
- End of definition phase: `call oasis_enddef (...)`
- Coupling field exchange:
 - in model time stepping loop
 - `call oasis_put (... , date , var_array. ...)`
 - `call oasis_get (... , date , var_array , ...)`
 - user's defined source or target (end-point communication)
 - sending or receiving at appropriate time only
 - automatic averaging/accumulation if requested
 - automatic writing of coupling restart file at end of run
- Termination: `call oasis_terminate (...)`

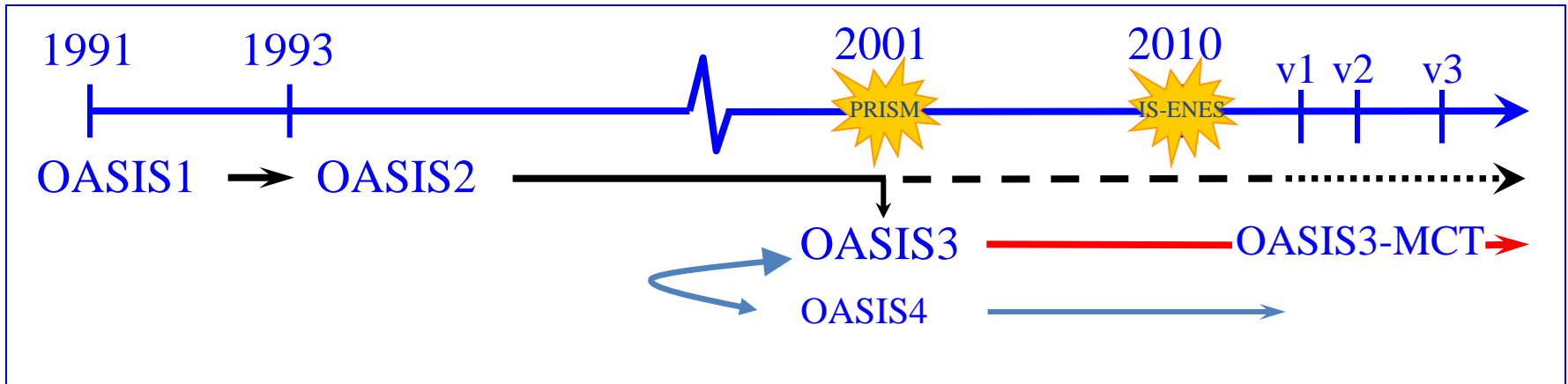


OASIS3-MCT: namcouple input file



** can now be generated with OASIS GUI based on CERFACS OPENTEA **

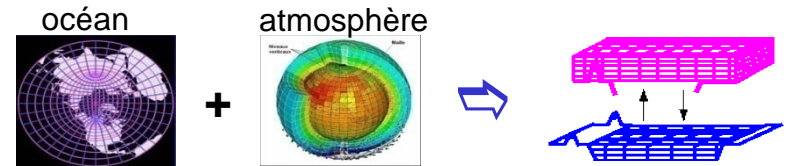
OASIS historical overview



• **OASIS1 -> OASIS2 -> OASIS3:**

2D ocean-atmosphere coupling
low resolution, low frequency

→ flexibility, modularity, 2D interpolations

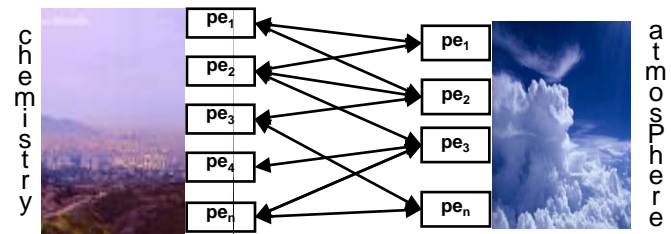


• **OASIS4 / OASIS3-MCT:**



2D/3D coupling of high resolution parallel components
on massively parallel platforms

→ parallelism, efficiency, performance



OASIS3-MCT_1.0 released July 2012
OASIS3-MCT_2.0 released May 2013
OASIS3-MCT_3.0 release imminent (May 2015)



OASIS3-MCT_3.0 New Features

- Ability to define grids, partitions and variables across and on subsets of processes between different executables or within the same executable
- Ability to couple on overlapping processes
 - component processes can overlap or partly overlap (in a single executable)
 - can couple fields within a component (ie. physics to dynamics) on same or different grids, same or different decompositions, same or different processes, etc.
 - user needs to avoid deadlocks, sends are non-blocking, receives are blocking, use namcouple lags to allow for 2-way coupling sequentially.
 - allows relatively arbitrary processor layout of components for science and load balance
- **Memory and Performance upgrades**
 - particularly in initialization phase
 - interaction with C. Goodyer, NAG, EXA2CT EU project
- New LUCIA load balancing tool and new memory tracking tool (gptl)
- Improved error checking and error messages
- Expanded test cases and testing automation
- Testing at high resolution (> 1M gridpoints), high processor counts (32k pes), and with large variable counts (> 1k coupling fields)
- Doxygen documentation





OASIS3 -> OASIS3-MCT -> OASIS3-MCT_3.0

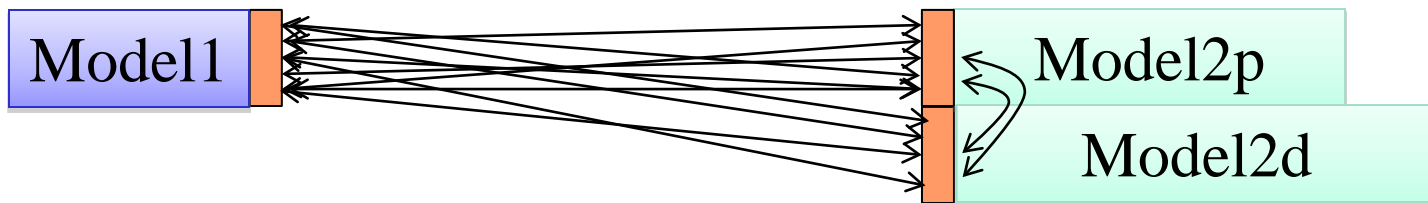
OASIS3



OASIS3-MCT_1.0

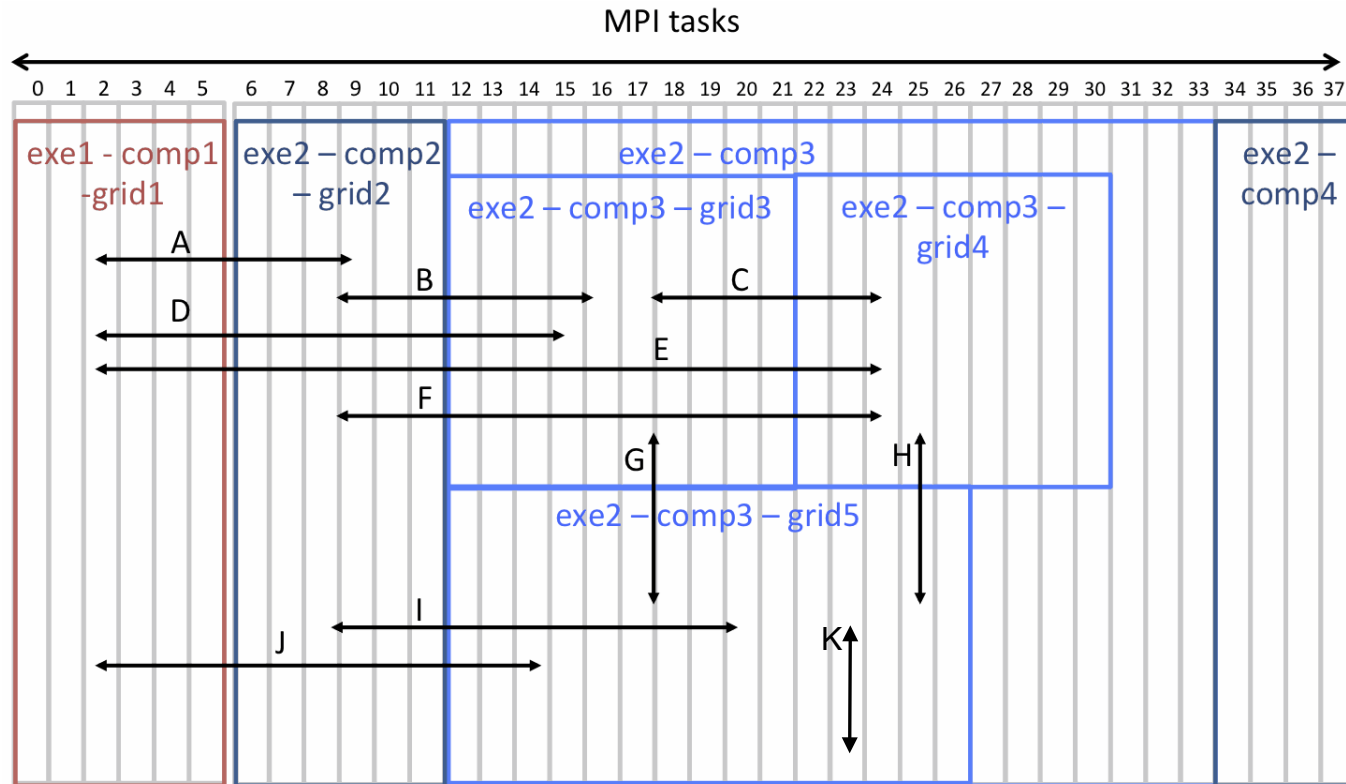


OASIS3-MCT_3.0





OASIS3-MCT_3.0 coupling capability

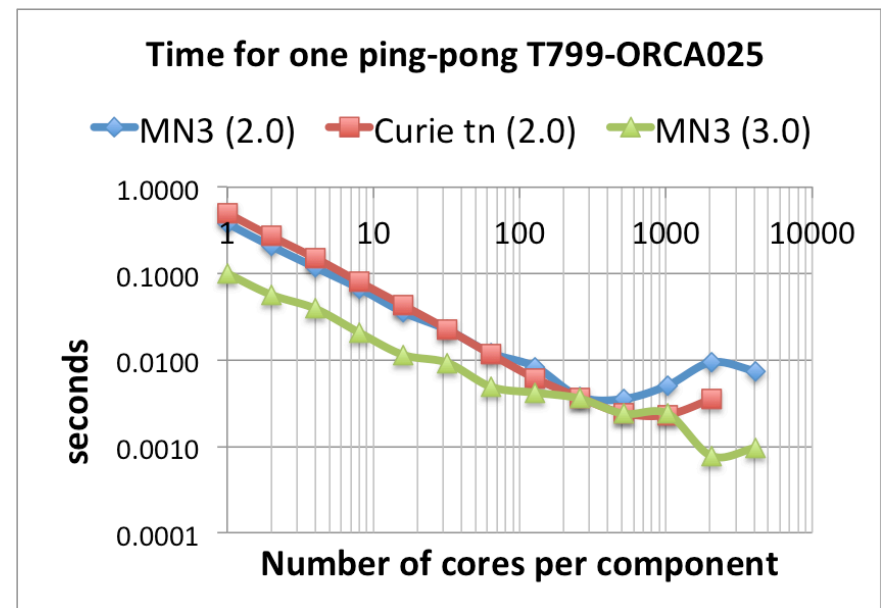
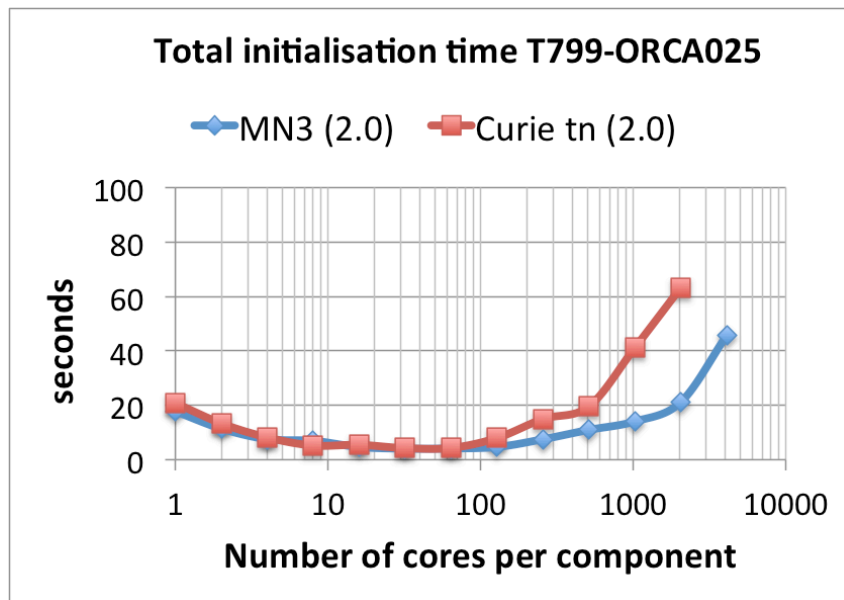


- System has 2 executables; `exe1` (atm) and `exe2` (ocn_ice)
- Executable 1 has 1 component and 1 grid (atm)
- Executable 2 has 3 components; `comp2` (ice), `comp3` (ocn), and `comp4`
 - `comp2` has 1 grid; `grid2` (ice) on all `comp2` processes
 - `comp3` has 3 grids (`ocn_phy`, `ocn_dyn`, `ocn_io`); on varying processes
- Supports many coupling combinations
- Prior to OASIS3-MCT_3.0, only coupling "A" was supported



OASIS3-MCT performance

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes; Intel® procs Sandy Bridge EP; IFort 12.1.7.256, Bullx MPI 1.1.16.5
- IBM MareNostrum3: Intel Sandy Bridge processors, Intel MPI 4.1.0.024

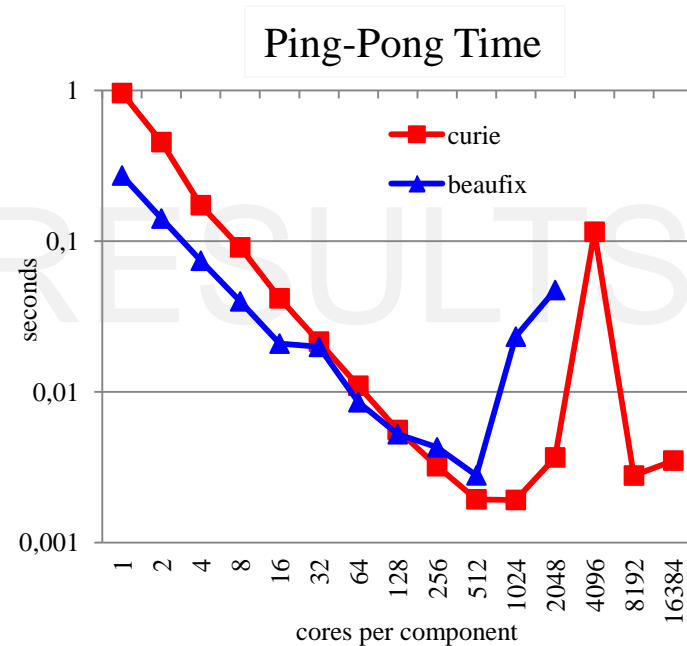
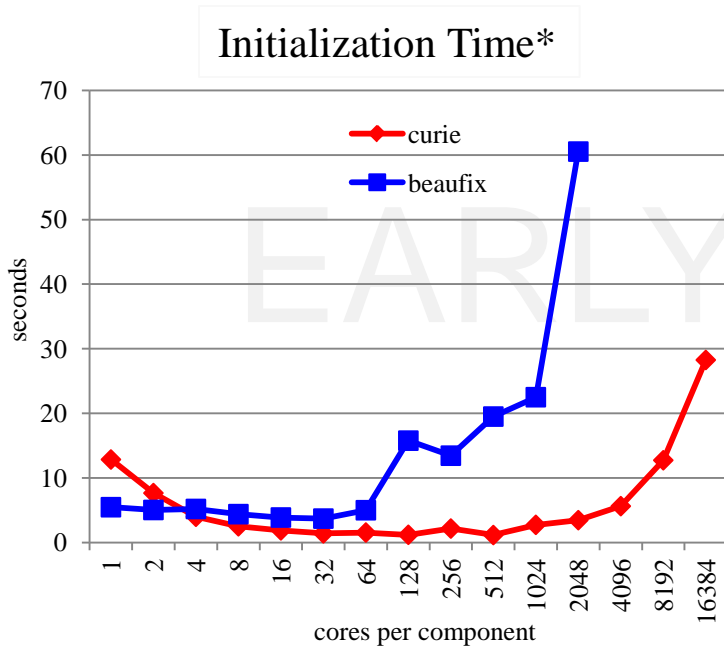




OASIS3-MCT performance (new results)

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes; Intel® procs Sandy Bridge EP; IFort 12.1.7.256, Bullx MPI 1.1.16.5
- Bullx Beaufix; Intel® Xeon Ivy Bridge; Infiniband FDR; IFort. Bullx MPI

- Results to 32k processes (16k processes for each component)



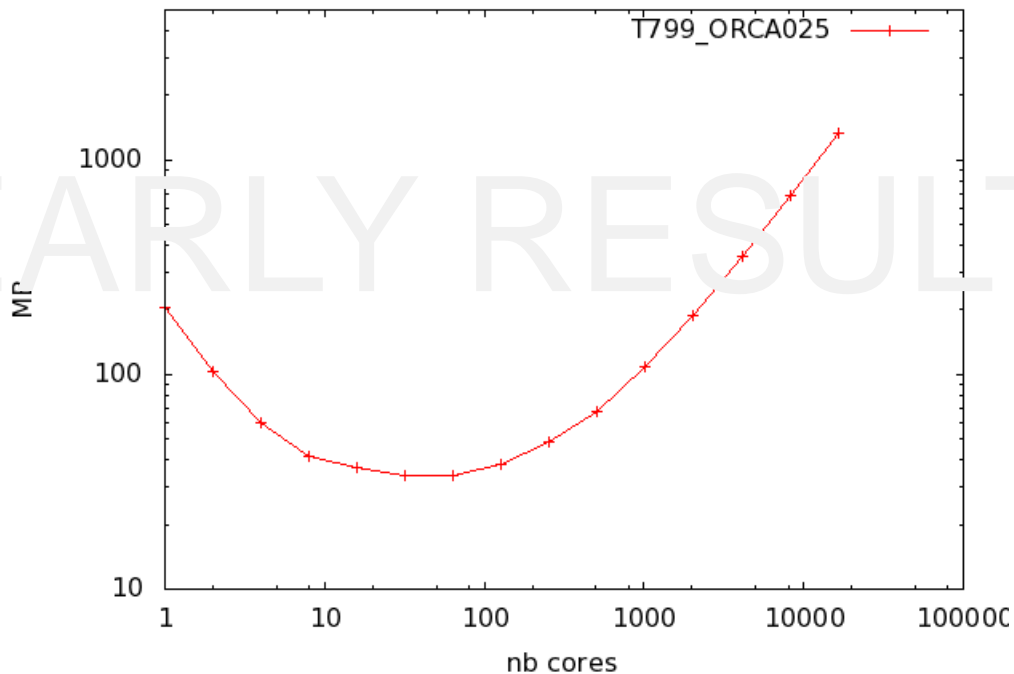
* cpl_setup



OASIS3-MCT Memory Use

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes; Intel® procs Sandy Bridge EP; IFort 12.1.7.256, Bullx MPI 1.1.16.5

Memory evolution for oasis_terminate (c2) T799_ORCA025 on Curie OASIS3-I



- Measured using the gptl tool
- MB/process



OASIS3-MCT_3.0 Summary

- OASIS3-MCT_3.0 release imminent (May 2015)
- Much greater flexibility to couple components
 - in a single executable
 - across disparate overlapping, non-overlapping, or partly overlapping processes
 - within a single component
- Better performance and more robust
- Significant effort made to tune and demonstrate viability at high resolution, high core counts, and large numbers of coupling fields.



OASIS3-MCT Future Work

- Features like "Bundled" fields (multiple 2d fields like multiple category ice)
- Continued evaluation of performance and scaling
- Review usage (ie. allow components to query oasis for coupling variables defined in namcouple)
- Continue evaluation of ESMF for off-line precomputing of interpolation weights
- IS-ENES2: Coupling technology benchmark + International Working Committee on Coupling Technologies (IWCCT, <http://earthsystemcog.org/projects/iwcct/>)
 - Performance of OASIS3-MCT for icosahedral grids
 - Evaluation of Open-PALM (including ONERA CWIPI library) and XIOS (IPSL I/O server)
 - Evaluation of ESMF



The end





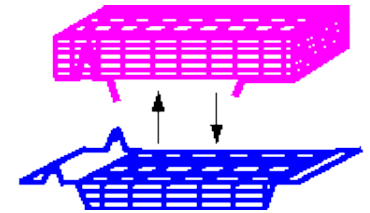
OASIS3-MCT, a coupler for climate modelling

S. Valcke, L. Coquart, CERFACS

- Introduction: Technical coupling solutions
- OASIS historical overview
- OASIS community today
- Use of OASIS3-MCT:
 - component model interfacing
 - coupled model configuration
- OASIS3-MCT communication
- OASIS3-MCT interpolations et transformations
- OASIS3-MCT performances
- Conclusions and perspectives

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?

- ✓ Coupling should be easy to implement, flexible, efficient, portable
- ✓ Coupling algorithm dictated by science (sequ. vs conc. coupling)
- ✓ Start from existing and independently developed codes
- ✓ Global performance and load balancing issues are crucial
- ✓ Platform characteristics (OS, CPU, message passing efficiency, ...)

1. merge the codes:

```
program prog1
...
call sub_prog2(data)
...
end prog1
```



```
program prog2
subroutine sub_prog2(data)
...
end prog2
```

- 😊 efficient (memory exchange)
- 😊 as portable as the codes
- 😊 one executable: easier to debug, easier for the OS
- 😊 sequential execution of the components

- 😞 not easy to implement with existing codes (splitting, conflicts in namespaces and I/O)
- 😞 not flexible (coupling algorithm hard coded)
- 😞 no use of generic transformations/interpolations
- 😞 loss of one degree of parallelism in the execution of the components

2. use existing communication protocols (MPI, CORBA, UNIX pipe, files, ...)

```
program prog1
...
call xxx_send (prog2, data, ...)
end
```

```
program prog2
...
call xxx_recv (prog1,data)
end
```

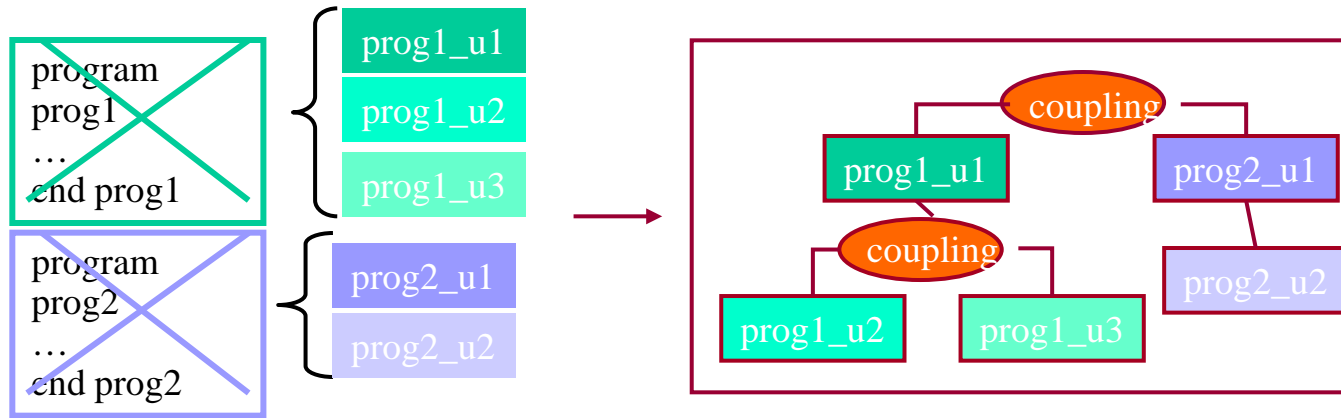
- 😊 existing codes
- 😊 natural parallelism in the execution of the components

- ☹ not easy to implement (needs protocol expert)
- ☹ not flexible (hard coded exchanges)
- ☹ multi-executable: possible waste of resources if forced sequential execution of the components
- ☹ multi-executable: more difficult to debug; harder to manage for the OS
- ☹ no use of generic transformations/interpolations
- ☹ efficient, portable

3. use coupling framework

ESMF **FMS**(GFDL) **CESM**(NCAR)

- Split code into elemental units
- Write or use coupling units
- Use the library to build a **hierarchical merged code**
- Adapt code data structure and calling interface

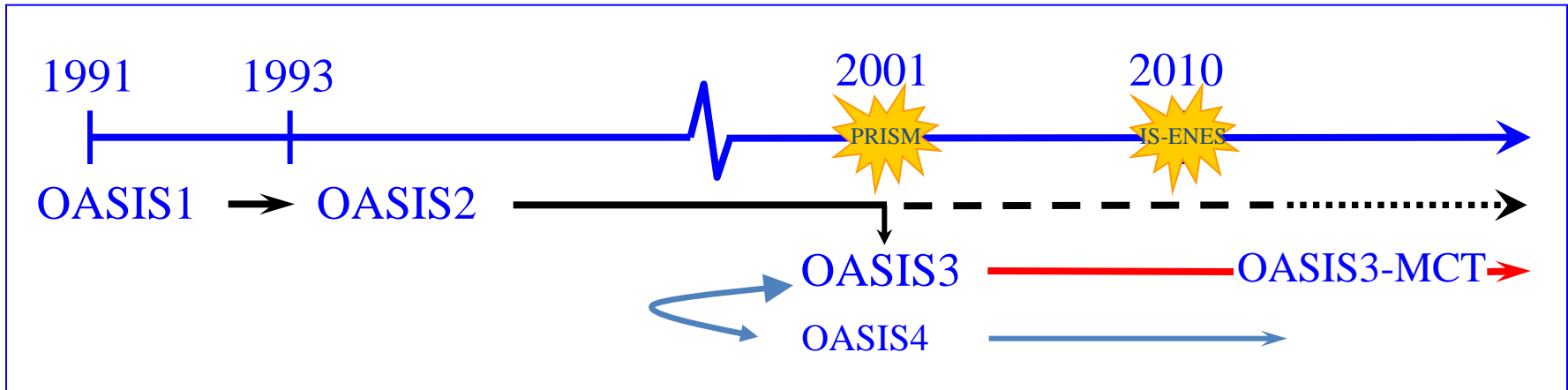


- ☺ efficient,
- ☺ sequential and concurrent components
- ☺ use of generic utilities (parallelisation, regridding, time management, etc.)

- ☹ existing codes
- ☹ (easy)

→ probably best solution in controlled development environment

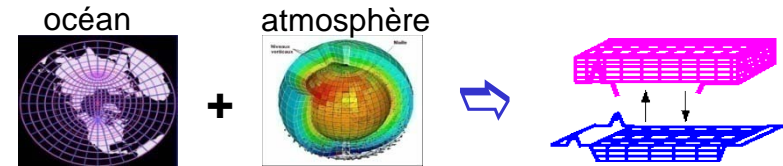
OASIS historical overview



- **OASIS1 -> OASIS2 -> OASIS3:**

2D ocean-atmosphere coupling
low resolution, low frequency

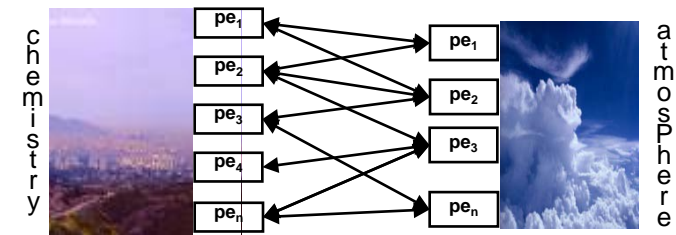
→ flexibility, modularity, 2D interpolations



- **OASIS4 / OASIS3-MCT:**

2D/3D coupling of high resolution parallel components
on massively parallel platforms

→ parallelism, efficiency, performance





OASIS community today

About 35 groups world-wide (climate modelling or operational monthly/seasonal forecasting):

- France: CERFACS, METEO-FRANCE, IPSL (LOCEAN, LMD, LSCE), OMP, LGGE, IFREMER
 - Europe: ECMWF + Ec-Earth community
 - Germany: MPI-M, IFM-GEOMAR, HZG, U. Frankfurt
 - UK: MetOffice, NCAS/U. Reading, ICL
 - Denmark: DMI
 - Norway: U. Bergen
 - Sweden: SMHI, U. Lund
 - Ireland: ICHEC, NUI Galway
 - The Netherland: KNMI
 - Switzerland: ETH Zurich
 - Italy: INGV, ENEA, CASPUR
 - Czech_Republic :CHMI
 - Spain: U. Castilla
 - Tunisia: Inst. Nat. Met
 - Japan: JMA, JAMSTEC
 - China: IAP-CAS, Met. Nat. Centre, SCSIO
 - Korea: KMA
 - Australia: CSIRO
 - New Zealand: NIWA
 - Canada: RPN-Environment Canada, UQAM
 - USA: Oregon State U., Hawaii U., JPL, MIT
 - Peru: IGP + downloads from Belgium, Nigeria, Colombia, Saudi Arabia, Singapore, Russia
- OASIS3 is used in 5 of the 7 European ESMs that participate in IPCC AR5
-

- All sources are written in F90 and C
- Uses the Model Coupling Toolkit (MCT) from Argonne National Lab
- Open source product distributed under a LGPL license
- All external libraries used are public domain (MPI, NetCDF) or open source (LANL SCRIP, MCT)
- Current developers are:
 - 1.3 permanent FTEs (CERFACS, CNRS)
 - 1 consultant (T. Craig, previously from NCAR)



IS-ENES (InfraStructure for ENES) EU FP7 project
2009-2012 - 18 partners - 7,6 MEuros ; coord: IPSL

➤ 93 pm for OASIS development and support



IS-ENES2, EU FP7 project 2013-2016 - 25 partners -
8MEuros

➤ 27 pm for OASIS3-MCT development and support



Use of OASIS3-MCT

At run time, the component models remain separate executables and OASIS3-MCT acts as a communication library linked to the models.

To use OASIS3-MCT:

- Download the sources, compile and run the tutorial on your platform
- Identify your component models, grids, coupling fields to be exchanged
- Identify the transformations to go from the source to the target grids
- Use the “test_interpolation” environment (offline) to test the quality
- Adapt your codes i.e. insert calls to OASIS3-MCT communication library
- Choose the other parameters (source and target, frequency, field transformations, etc.) and create the *namcouple* configuration file with the GUI
- Compile OASIS3-MCT, your components **with same compiler**, and link the components models with OASIS3-MCT library
- Start the models and let OASIS3-MCT manage the coupling exchanges



OASIS3-MCT: component model interfacing

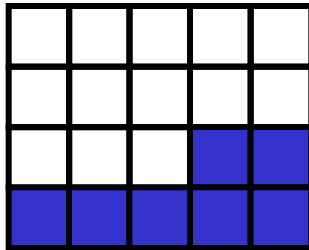
- Initialization: `call oasis_init_comp(...)`
- Grid definition: `call oasis_write_grid (...)`
- Local partition definition: `call oasis_def_partition (...)`
- Coupling field declaration: `call oasis_def_var (...)`
- End of definition phase: `call oasis_enddef (...)`
- Coupling field exchange:
 - in model time stepping loop
 - `call oasis_put (... , date , var_array. ...)`
 - `call oasis_get (... , date , var_array , ...)`
 - user's defined source or target (end-point communication)
 - sending or receiving at appropriate time only
 - automatic averaging/accumulation if requested
 - automatic writing of coupling restart file at end of run
- Termination: `call oasis_terminate (...)`



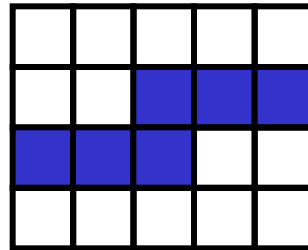
OASIS3-MCT: partitioning supported

Apple

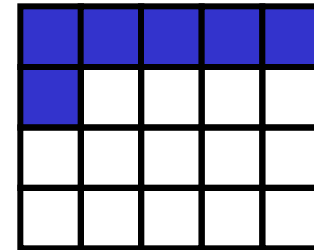
part1



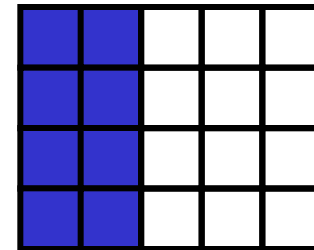
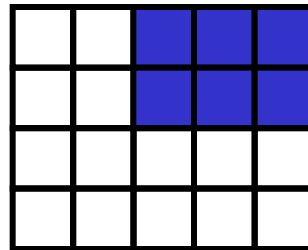
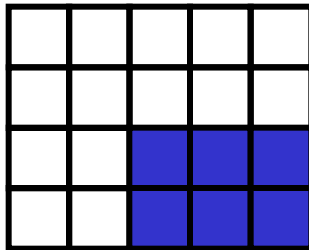
part2



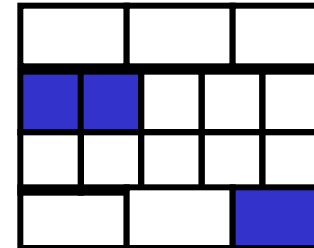
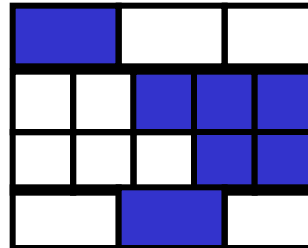
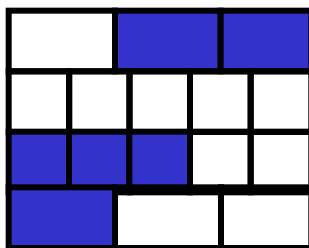
part3



Box



Orange



Apple and orange applicable to unstructured grids



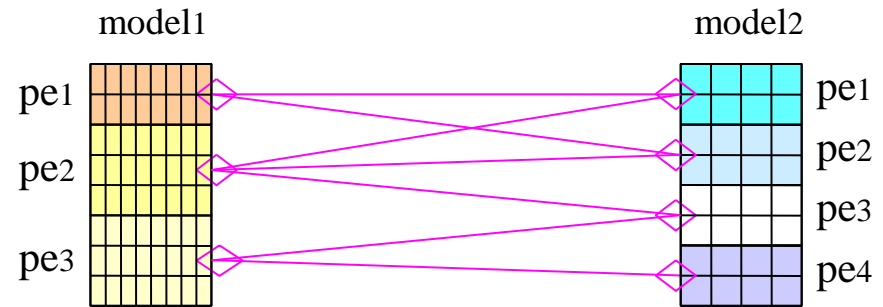
OASIS3-MCT coupled model configuration

Configuration in a **text** file *namcouple*

** now generated with OASIS GUI based on CERFACS OPENTEA **

- general characteristics of a coupled run
 - total duration
 - components
 - ...
- for each exchange of coupling field :
 - source and target symbolic name (end-point communication)
 - exchange period
 - transformations/interpolations

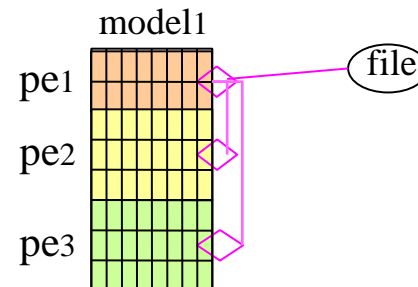
- Fully parallel communication between parallel models based on Message Passing Interface (MPI)



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: interpolations & transformations

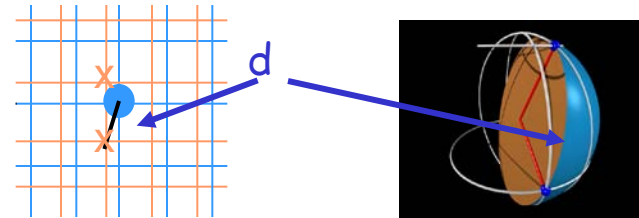
- on 2D or 3D scalar fields
- on different types of grids: lat-lon, rotated (logically rectangular), gaussian reduced, unstructured

❖ Transformations: statistics, addition/multiplication by scalar, global conservation

❖ Interpolations/regridding SCRIP (Jones, 1999)

n-nearest-(gaussian-weighted)-neighbours: $\text{weight}(x) \propto 1/d$

d: great circle distance on the sphere:

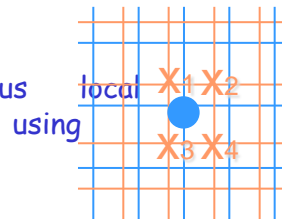


x source grid point
● target grid point

bilinear interpolation

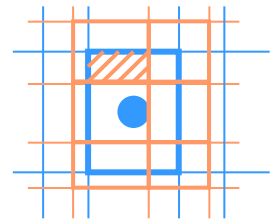
➤ general bilinear iteration in a continuous coordinate system

$f(x)$ at x_1, x_2, x_3, x_4



conservative remapping

➤ weight of a source cell % to intersected area



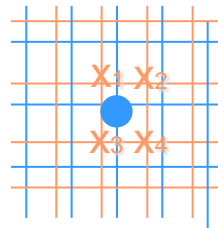
bicubic interpolation:

➤ general bicubic iterations in a continuous local coordinate system:

$f(x), \delta f(x)/\delta i, \delta f(x)/\delta j, \delta^2 f/\delta i \delta j$ in

x_1, x_2, x_3, x_4

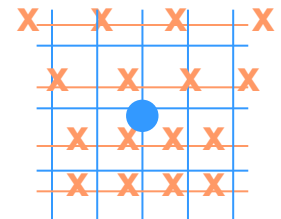
for logically-rectangular grids (i,j)



➤ standard bicubic algorithm:

16 neighbour points

for Gaussian Reduced grids

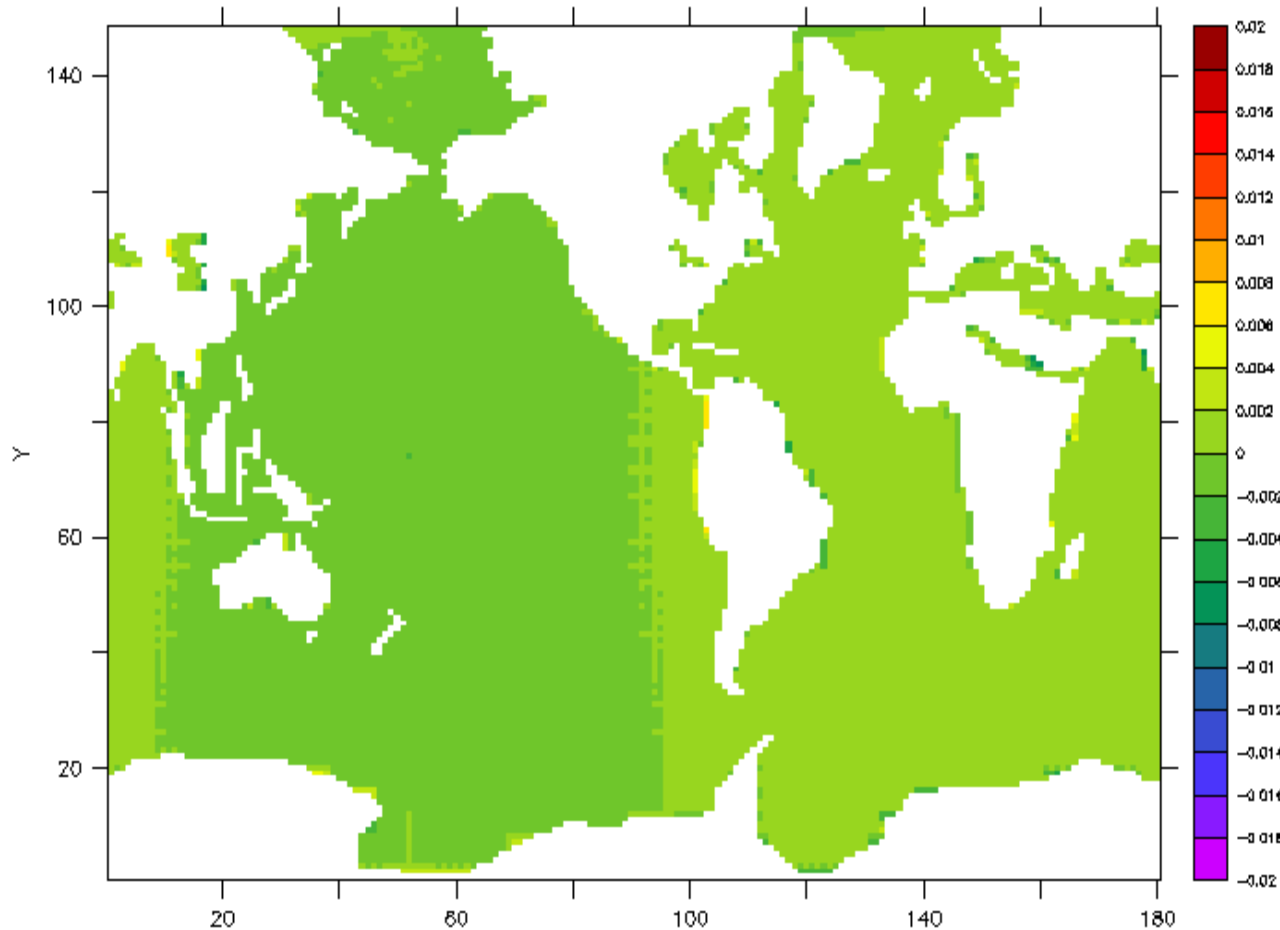
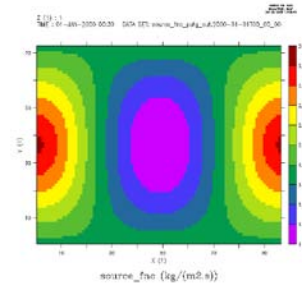


** gradients must be given as extra arguments to the oasis_put

One example of bilinear interpolation error

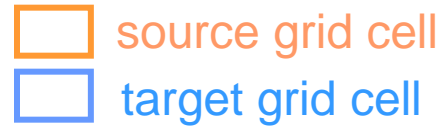
$$F = 2 + \cos[\pi * \text{acos}(\cos(\text{lon})\cos(\text{lat}))]$$

LMDz grid (96 x 72) → ORCA2



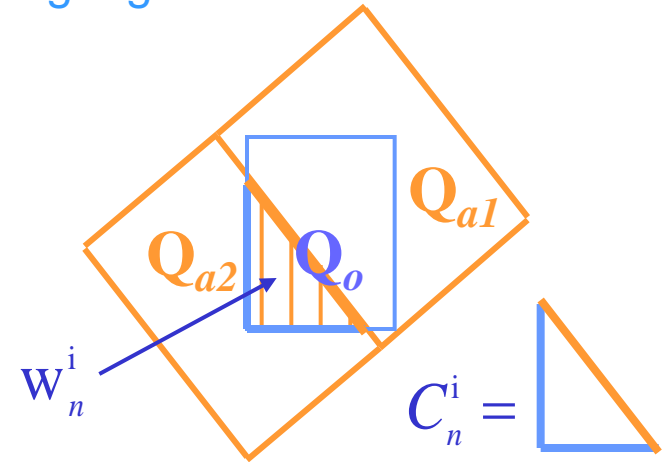
➤ < 0.2% whole domain; ~1% near the coastline

1st order conservative regridding:



- conserves integral of extensive properties
- weight of a source cell α to intersected area

$$Q_o^i = \frac{1}{A_o} \sum_{n=1}^N Q_{a_n} w_n^i \quad \text{with} \quad w_n^i = \oint_{C_n^i} -\sin(\text{lat}) d\text{lon}$$



Actual limitations:

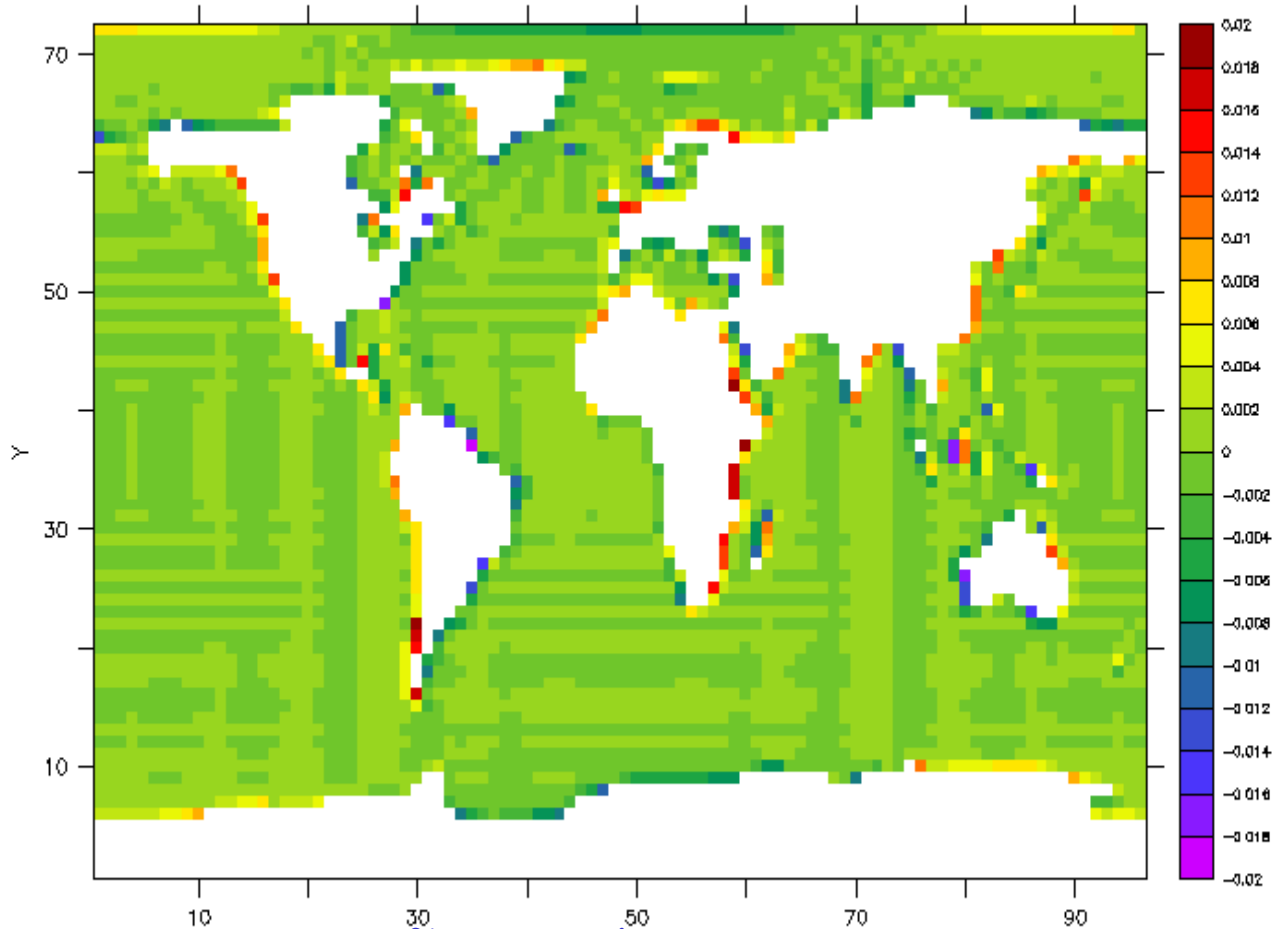
- assumes borders are linear in (lat,lon) ; uses Lambert equivalent azimuthal projection near the pole for intersection calculation
 - assumes $\sin(\text{lat})$ linear function of lon for line integral calculation
 - ❖ need to use a projection near the pole (as done for intersect. calc.)
 - exact calculation is not possible as "real shape" of the borders are not known
 - ❖ could use of border middle point
 - ❖ to ensure conservation, need to normalize by true area of the cells (under work)
- ESMF new conservative regridding under evaluation (see examples/test_rmp_esmf)

OASIS3-MCT: interpolations & transformations

- One example of conservative remapping error

$$F = 2 - \cos[\pi * \text{acos}(\cos(\text{lon})\cos(\text{lat}))]$$

ORCA2 → LMDz (96x72)



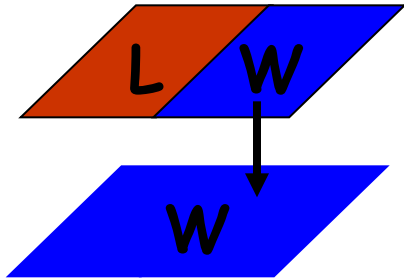
- < 0.2% everywhere except
- ~ 0.8% for LMDz last row close to the North pole
- ~ 2% near the coastline

Problem with non-matching sea-land masks

$$Q_o^i = \frac{1}{A_o} \sum_{n=1}^N Q_{an} W_n^i$$

1- Support subsurfaces in the atmosphere

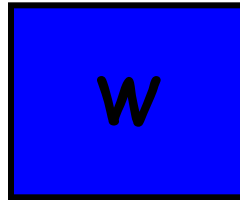
and use the ocean land-sea mask in the atmosphere to determine the fractional area of each type of surface



2-“DESTAREA” option

- local flux conservation
- possibly unrealistic flux values

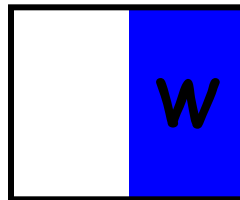
$$A_o =$$



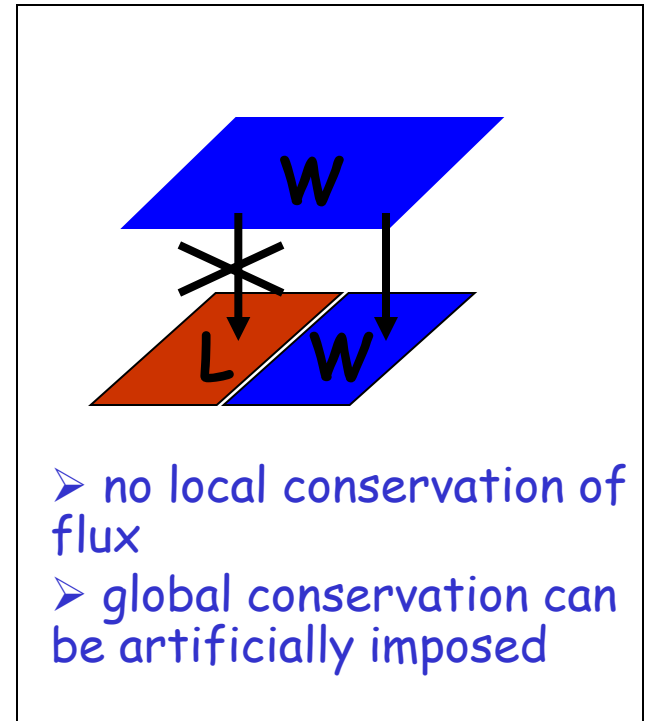
3-“FRACAREA” option

- no local conservation of flux
- realistic flux values

$$A_o =$$



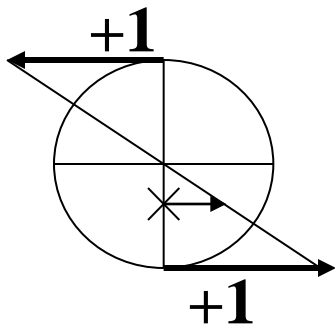
+ nearest non-masked value for ocean cells covered only with masked atmospheric cells (FRACNNEI)



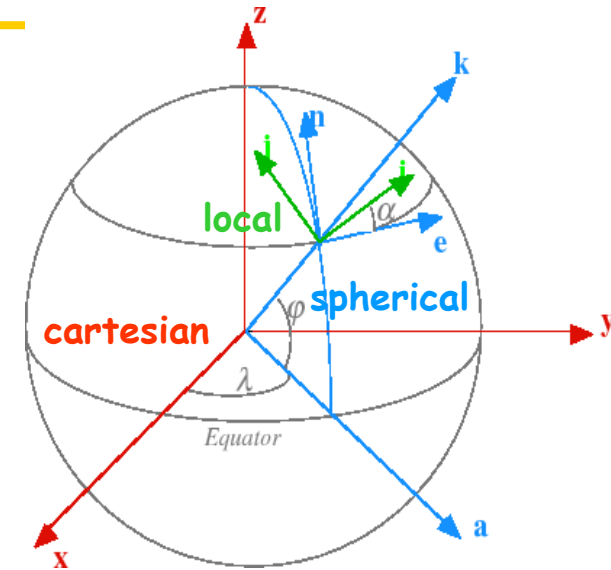
Vector interpolation (winds, currents, ...)

- ❖ interpolation of vectors component per component is not accurate, especially where the referential changes rapidly

Example interpolation of a zonal wind in the spherical referential near the pole



- At x, one would expect a zonal wind between 0 and 1.
- Interpolation comp. per comp. → zonal wind of 1.



Solution (proposed by O. Marti, LSCE):

- “turn” the vector in the spherical ref. and project the resulting vector in a cartesian ref
- send the 3 components in the cartesian referential and let OASIS3-MCT interpolate them
- project back in spherical referential; check that k component is zero
- possibly “turn” the resulting vector in the target local referential



OASIS3-MCT users

- ◆ CERFACS (France):
 - NEMO ocean (ORCA025, 1021x1442) - ARPEGE atmosphere (Gaussian Red T359 grid, 181724 points).
 - Seasonal prediction exp, SPRUCE PRACE project, 27 Mhours, Bullx Curie.
 - Decadal prediction exp, SPECS PRACE project, 10 Mhours, MareNostrum3

- ◆ IPSL (France):
 - WRF atm - NEMO, 2-way nested zooms, 27-9 km (4322x1248 pts), 7 cpl fields, 1h cpl period.
 - PULSATION project (ANR), 22 Mhours on PRACE tiers-0 Bullx Curie.

- ◆ MPI-M (Germany):
 - MPI-ESM-XR: atmosphere ECHAM6 T255L95 (768x384 grid pts, ~50km) - ocean MPIOM TP6ML40 (3602x2394 grid pts, ~10km); 17 cpl fields, 1h cpl period.

- ◆ MetOffice (UK) :
 - UM global atm (N768, 1536x1152) - NEMO (ORCA012, 4320x3058), 38 cpl fields, 1h-3h cpl frequency

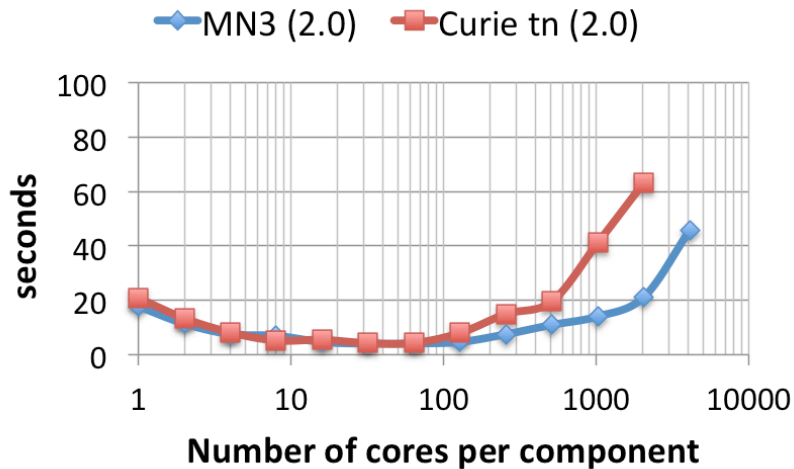
- ◆ BTU-Cottbus (Germany)
 - 3D coupling: COSMO-CLM regional atm (221x111x47, ~2 deg) - ECHAM global atm (T63, 192x96x47), + 2D coupling to MPI-OM ocean (254x220)
 - 6% coupling overhead observed for exchange of 6 x 3D fields every ECHAM time step

- ◆ ... + many others (NICAM-NEMO, EC-Earth, ...)

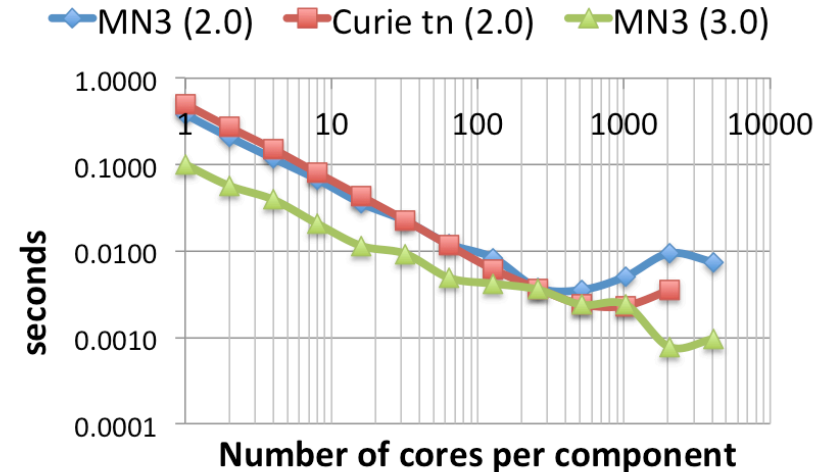
OASIS3-MCT performance

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes; Intel® procs Sandy Bridge EP; IFort 12.1.7.256, Bullx MPI 1.1.16.5
- IBM MareNostrum3: Intel Sandy Bridge processors, Intel MPI 4.1.0.024

Total initialisation time T799-ORCA025



Time for one ping-pong T799-ORCA025



Coupling overhead for one-year long simulation with one 1 coupling exchange every hour in each direction between codes with $O(1\text{ M})$ grid points running on 4000 cores/component:

- ~20 seconds for initialisation, ~9 seconds for data exchange



Conclusions and perspectives

Conclusions on OASIS3-MCT

- Good performance, removes OASIS3.3 bottleneck
 - Very simple to use for traditional OASIS3.3 users (same API)
- > OASIS3-MCT most likely provides a satisfactory solution for fully parallel coupling in our climate models at the resolutions targeted operationally for the next ~5 years.

Perspectives:

- Release of OASIS3-MCT_3.0 in January 2015: more flexibility in coupling layout
- Evaluation of ESMF for off-line precomputing of interpolation weights (on going)
- IS-ENES2: Coupling technology benchmark + International Working Committee on Coupling Technologies (IWCCT, <http://earthsystemcog.org/projects/iwcct/>)
 - Performance of OASIS3-MCT for icosahedral grids
 - Evaluation of Open-PALM (including ONERA CWIPI library)
 - Evaluation of ESMF



The end