IS-ENES – WP 7

D7.2 - Fully parallelised and optimised version of OASIS4 answering needs of current coupled climate models

Abstract:

This document reports on the work done for the development of the OASIS4 coupler. A general description and results of scalability tests performed with the latest OASIS4 version, OASIS4_1beta, are first reported. All developments, bug fixes and validation tests done in IS-ENES are detailed and known problems are listed.

However, even if the efforts devoted since the beginning of the project are already more important than the total planned, we conclude here that we are still not able to deliver a stable version of OASIS4 parallel neighbourhood search library and that this objective is out of reach given the current resources.

Fortunately, as the OASIS3 version of the coupler still answers the needs of current coupled climate models, this conclusion does not constitute a blocking issue. For the next steps, we propose to focus on the OASIS4 user-defined regridding functionality, which bypasses the parallel neighbourhood search while ensuring the parallel redistribution of the source coupling data directly between the source and the target processes.

Grant Agreement Number:	**228203**	**Proposal Number:**	**FP7-INFRA-2008-1.1.2.21**
Project Acronym:	**IS-ENES**		
Project Co-ordinator:	**Dr Sylvie JOUSSAUME**		

Document Title:	Fully parallelised and optimised version of OASIS4 answering needs of current coupled climate models			**Deliverable:**	D 7.2
Document Id N°:		**Version:**	0.3	**Date:**	October 21, 2011
Status:	Final				
Filename:	ISENES_Deliverables_7.2.docx				
Project Classification:	Public				

Approval Status		
Document Manager	**Verification Authority**	**Project Approval**

REVISION TABLE

Version	Date	Modified Pages	Modified Sections	Comments
0.1	2011/07/27			First version by S. Valcke
0.2	2011/09/20			Internal review by G. Riley, G. Aloisio, R. Hill, M.-A. Foujols
0.3	2011/10/19			Final review by S. Valcke

TABLE OF CONTENTS

Executive Summary

The objective of Task 2 “OASIS4 development” of IS-ENES WP7/JRA1 was to deliver a fully parallelised and optimized coupler offering 2D/3D linear and cubic interpolations and 2D conservative remapping for current coupled climate models.

The current document reports on the work done to reach this objective. The latest OASIS4 version, OASIS4_1beta, is first described. In particular, OASIS4_1beta includes parallel interpolation for Gaussian Reduced grids and 2D conservative remapping. We also conclude that the first scalability tests are encouraging and can be used as a proof-of-concept.

However, although the efforts devoted since the beginning of IS-ENES for these developments and related validation are already more than the total planned for the whole project. Some problems that are mainly in the parallel neighbourhood search still remain. We therefore conclude here that the original objective, which was to deliver a fully operational coupler performing online parallel calculation of the interpolation weights and addresses, is out of reach given the current resources.

Fortunately, we also show that the OASIS3 version of the coupler still answers the needs of current coupled climate models. However, as its limited field-per-field parallelism will most probably become a bottleneck for higher resolution simulations, we propose here to focus on the OASIS4 user-defined regridding functionality, which bypasses its parallel neighbourhood search. This coupling solution should allow us to efficiently run the next climate models on massively parallel platforms.

1. Introduction

The objective of Task 2 “OASIS4 development” of IS-ENES WP7/JRA1 was to deliver a fully parallelised and optimized coupler offering 2D/3D linear and cubic interpolations and 2D conservative remapping for current coupled climate models. To achieve this task, the Description of Work (DoW) proposed four main developments:

- validation of the parallel global search for Gaussian Reduced grids
- implementation of the global parallel search for the 2D conservative remapping
- possibility of running more than one component models sequentially within one executable
- support of regional model specificities

Assessment of OASIS4 performance and scalability was also mentioned in the DoW. On the longer term, the main issue that came out of the User Survey (WP7/JRA1 Task 1), done to help prioritise the longer-term developments, was the support of unstructured grids.

The current document reports on the work done to reach these objectives. Section 2.1 provides a general description of the current OASIS4 version, OASIS4_1beta, while section 2.2 reports on first scalability and performance tests done for the multi-grid neighbourhood search and for the Driver/Transformer (D&T). In section 2.3, we detail the developments and bug fixes done in IS-ENES; this section presents in particular, the work done to address the first three developments listed above. The validation tests based on toy models and offline regridding quality analysis are described in section 2.4, while known remaining problems are listed in section 2.5. In section 3, we discuss the current status of OASIS4 and we propose concrete actions for the next steps. Section 4 proposes a summary of the document.

2. OASIS4_1beta

In 1991, CERFACS started research in climate modelling with the objective to perform the technical assembling of an ocean General Circulation Model (GCM), OPA, and two different French atmospheric GCMs, ARPEGE and LMDz. After an initial period of investigation, it was decided that the technical coupling layer between the ocean and atmosphere components should take the form of an external coupler, i.e. a separate executable performing the regridding of the coupling fields and a coupling library linked to the components performing the coupling exchanges as defined by the user in an external configuration file. This choice ensured a minimal level of interference in the existing codes while focussing on modularity and portability. As the coupling was, at the time, involving only a relatively small number of 2D coupling fields at the air-sea interface, efficiency was not considered a major criterion. Two years later, a first version of the OASIS coupler was distributed to the community. From 2001 until 2004, the development of OASIS benefited from an important support from the European Commission in the framework of the PRISM project [Valcke2006a]. Collaboration with NEC Laboratories Europe - IT Research Division (NLE-IT), SGI and the French Centre National de la Recherche Scientifique (CNRS) originated during that period. During PRISM, the OASIS3 version, the direct evolution of the OASIS coupler developed since 1991 at CERFACS, was released. As the climate modelling community is progressively targeting higher resolution climate simulations run on massively parallel platforms with coupling exchanges involving a higher number of (possibly 3D) coupling fields at a higher coupling frequency, the development of a new fully parallel coupler, OASIS4, also started during PRISM. Parallelism and efficiency drove OASIS4 developments, at the same time keeping in its design goals; the concepts of portability and flexibility that made the success of OASIS3.

2.1 General description

The Fortran and C routines that constitute OASIS4 form, after compilation, a separate Driver & Transformer executable (D&T) performing driving and regridding tasks and a model interface library, the PSMILe, that needs to be linked to and used by the component models. Functions supported in the current version OASIS4_1beta are described hereafter. Its sources are available at https://oasistrac.cerfacs.fr/browser/tags/OASIS4_1beta .

2.1.1 Coupling configuration

At run time, OASIS4 D&T reads the coupled run configuration defined by the user before the run and distributes the corresponding information to the different component model PSMILes. This user-defined configuration, provided in Extensible Markup Language[1] (XML) files, contains all coupling options for a particular coupled run, e.g. the duration of the run, the component models, and for each coupling exchange a symbolic description of the source and target, the exchange period, regridding and other transformations. A Graphical User Interface (GUI) facilitates the creation of those XML files. During the run, the D&T executable and the component model PSMILes perform appropriate exchanges based on this configuration.

2.1.2 Process management

In a coupled run using OASIS4, the component models remain as separate executables with their main characteristics, such the general code structure or the memory management, unchanged with respect to the standalone mode. The user has to ensure that the component models coherently define some global parameters such as the total run duration, the calendar, etc.

OASIS4 supports two ways of starting the executables of the coupled application. If a complete implementation of the MPI2 standard [Gropp1998] is available, only the OASIS4 D&T has to be started by the user. All remaining component executables are then launched by the OASIS4 D&T at the beginning of the run using the MPI2 MPI_Comm_Spawn functionality. If only MPI1 [Snir1998] is available, the OASIS4 D&T and the component model executables must be all started at once in the job script in a "multiple program multiple data" (MPMD) mode. The advantage of the MPI2 approach is that each component keeps its own internal communication context unchanged with respect to the standalone mode, whereas in the MPI1 approach, OASIS4 needs to recreate a component model communicator that must be used by the component model for its own internal parallelisation. However, the drawback of the MPI2 “spawn” approach is that it is not universally suitable for all batch schedulers. In both cases, all component models are necessarily integrated from the beginning to the end of the run, and each coupling field is exchanged at a fixed frequency defined in the configuration file for the whole run.

2.1.3 Communication: the OASIS4 PSMILe library

To communicate with other component models or to perform I/O, a component model needs to call few specific OASIS4 PSMILe routines. The PSMILe API function calls can be split into three phases. The first phase includes calls for the coupling initialisation, the definition of the grids (i.e. the grid point and corner longitude and latitude), the description of the local partition in a global index space, and the coupling field declaration; the second phase comprises receiving and sending of the coupling fields (by calling respectively a prism_get or a prism_put routine) usually implemented in the model timestepping loop, while the third phase terminates the coupling.

The sending and receiving of data is managed by the PSMILe below the prism_get and prism_put calls, following a principle of “end-point" data exchange. When producing data, no assumption is made in the source component code concerning which other component will consume these data or whether they will be written to a file, and at which frequency; likewise, when asking for data, a target component does not know which other component model produces them or whether they are read in from a file. The target or the source (another component model or a file) for each field is defined by the user in the configuration file and the coupling exchanges and/or the I/O actions take place according to the user external specifications. This implies in particular that the switch between the coupled mode and the forced mode is totally transparent for the component model. MPI is used for coupling exchanges, while I/O actions are based on GFDL mpp_io library [Balaji2001]. Furthermore, the prism_get and prism_put routines can be placed anywhere in the

[1] http://www.w3.org/XML/

source and target code and possibly at different locations for the different coupling fields. These routines can be called by the model at each timestep. The actual date at which the call is valid is given as an argument. The sending/receiving is only performed when the date corresponds to the times at which it should be activated, indicated by the field coupling or I/O frequency as set by the user in the configuration file; a change in the coupling or I/O frequency is therefore also totally transparent to the component model itself.

OASIS4 PSMILe supports parallel communication in the sense that each process of a parallel model can send or receive its local part of the field. The communication pattern between the source and target component processes is based on the intersection of the local domains covered by each source and target process and on the needs of the regridding between the source and target grids. The parallel neighbourhood search at the base of the regridding and therefore at the base of the communication patterns between the source and target processes is detailed in [Redler 2010] and briefly explained in the next paragraphs.

2.1.4 OASIS4 global parallel neighbourhood search

i. General algorithm for block-structured grids

The OASIS4 PSMILe library performs the exchanges of coupling data between source and target components. Usually those components express their coupling fields on different numerical grids and a regridding operation has to be performed on the coupling data. The regridding implies:

1. identifying the ``neighbours'' of each target point, i.e. the source grid points that will contribute to the calculation of the target grid point value in the regridding process (the ``neighbourhood search''),
2. calculating the weights of the different neighbours,
3. performing the calculation of the grid point values.

To minimize the transfer of source data, it was decided to perform the neighbour search (1.) in the source PSMILe and to transfer only the useful source grid points to the D&T, which performs the regridding calculation per se (i.e. 2. and 3.).

In an initial step, each component process determines the envelope of its locally defined grid partition. The envelopes are exchanged between those component processes that have to exchange data with each other. Pairs of source and target processes that have common intersections are identified. For each of these intersections, lists of target grid points included in the intersection are generated on the target side and transmitted to the respective source process. Only the envelopes of all partitions are stored on all processes; for all steps in the PSMILe or in the D&T, it is never required to gather the global grid information at a central place onto one single process.

The next stage of the search is to find a reference source point for each target point, which can later be used as a starting point to build the required regridding stencil, i.e. the set of source neighbour points used for the calculation of the target point value. On the source side, a block-structured grid hierarchy is established out of the local source grid. This grid hierarchy, illustrated on Figure 1, is constructed by splitting the local source grid into smaller subsections with a refinement factor of two; the subsection on the highest (finest) level in the grid hierarchy typically contains three grid points in each direction.

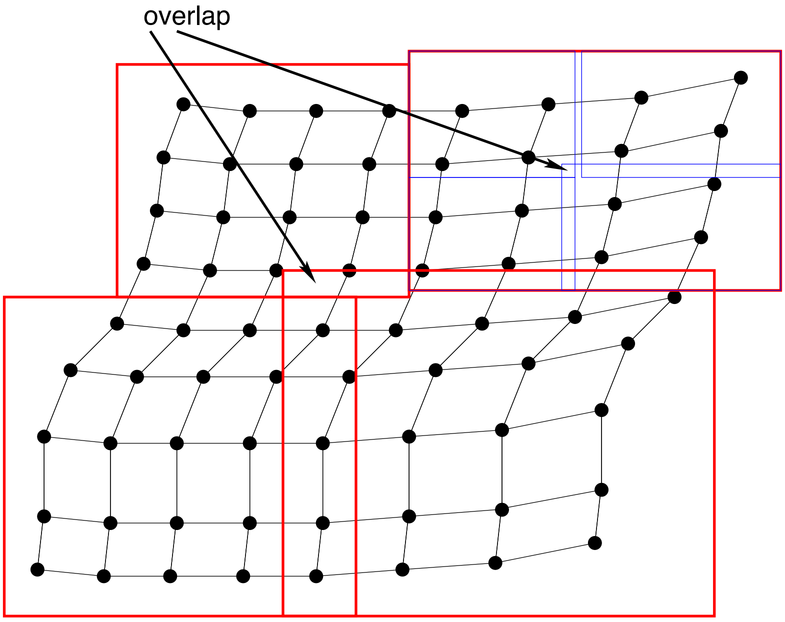

Figure 1 – Illustration of a grid hierarchy used in the multigrid search

A great advantage of the multi-grid algorithm is that it is only weakly dependent on the problem size: when the grid size is doubled in each horizontal direction this introduces only one more multi-grid level. The performance of the OASIS4 multi-grid search is further discussed and compared with the performance of a classical search in section 2.2.3 below. Once the containing source cell is identified, the ``lower-left'' source-point (in the [i,j] index space) is determined and stored as the reference source point for the particular target point.

In the final step, the remaining source neighbours required to perform the regridding are identified. For example, for a bilinear interpolation, the four enclosing source points will be identified as the four neighbour source points needed.

For some target points, some source points first identified as neighbour points may not be available for the regridding if they are masked out, which means that the physical fields do not necessarily contain meaningful values on those points. In such cases the user has different choices via the XML configuration file. As a default option, the remaining valid points will be used for a weighted-distance interpolation. Another option available is to not interpolate onto such target points. In the worst case when all neighbour source points identified by the default search are masked, OASIS4 offers the possibility to search for the non-masked nearest-neighbour value and use this value for the target point. For this extra nearest-neighbour search, the algorithm starts from the original masked source grid point taken as an initial guess. Starting from the finest level containing the initial guess, the multi-grid hierarchy is then used again to locate the closest non-masked source points by moving in so-called w-cycles through the multi-grid hierarchy, that is going down to a coarser level and going up again in another branch of the hierarchy repeatedly until the closest point is found.

On modern parallel systems, the component domain is usually distributed to many processes and/or processors. If the regridding of fields was performed using only grid points available on the actual process, the result of the regridding would depend on the actual partitioning. For example, if a target point is located close to the internal border of a partitioned domain of the source application, the standard regridding stencil may require source points which are not located on the actual process but on a remote neighbour process. The PSMILe search, called ``global search'', considers remote neighbour points appropriately, and therefore ensures that the regridding is invariant to the domain partitioning. Since the full connectivity information for partitioned grids is not provided to the OASIS4 PSMIle library, missing source points for regridding stencils require an additional search step on remote neighbouring domains. When these remote source neighbour points are found, the full information including the mask data are returned to the process that has initiated this additional search step. At run-time, this process will collect the data from possibly different processes and will forward the data for the full regridding to the D&T.

ii. Neighbourhood search for Gauss-reduced grids

Gauss-reduced grids have been introduced mainly for atmospheric models, in particular the ARPEGE and IFS models, to overcome problems resulting from the convergence of meridians close to the geographical poles. These grids cannot be classified as block-structured grids, nor are they completely unstructured. In this particular case, some implicit knowledge about how the grid is constructed can be used. In order to reuse as many of the existing functions as possible, a block-structured auxiliary grid, regular in longitude and latitude, is set up and the corresponding mapping of the Gauss-reduced grid points onto this auxiliary grid is determined. The initial search on the auxiliary grid and the identification of the initial locations can now be performed. By mapping them back onto the Gauss-reduced grid, the correct start location is obtained and the local search is pursued on the Gauss-reduced source grid in a similar fashion as it has already been described for block-structured grids. In contrast to block-structured grids, the connectivity cannot be derived from the i-j indices directly. Instead the connectivity among the grid points stored in an 1-D array is determined based on the description of the Gauss-reduced grid provided by calling the routine PRISM_Reducedgrid_Map in the component code.

iii. Cell-based search for conservative remapping

Compared to the search described above for point-based interpolation schemes, the cell-based search used for the conservative remapping has to follow a slightly different approach. The task here is to locate the overlapping source cells for each target cell. A point-based search is first performed on the lower-left corner of each target cell in a very similar way to the search described above. Once the source cell containing the lower-left corner of a target cell is identified, the local search can be continued to identify the remaining source cells which potentially overlap a given target cell. The overlap between a source and a target cell is determined by a pairwise investigation of the intersection of the edges without a detailed evaluation of the exact area of overlap.

2.1.5 PSMILe-Driver/Transformer interaction and regridding

The OASIS4 PSMILe supports two different ways of exchanging data: when the source and target grid points match, no regridding is needed and the data is exchanged directly between two components; but when regridding is required, the coupling data is exchanged via the D&T processes where the regridding is performed.

As the final result of the PSMILe search described in the last section, each source process holds 1-dimensional lists, each list containing the geographical information relevant for the regridding of the intersection of the local source grid domain with one target process grid domain, the ``intersection regridding lists''. These intersection regridding lists are then equally distributed over the D&T processes available. This ensures that the D&T processes work in parallel on the regridding of the different source and target process intersections.

During the run, the different D&T processes wait for messages to arrive from any component PSMILe, receive header messages containing information about the next message to receive or pending requests to fulfil, and perform related actions. They repeat this sequence of actions in an indefinite loop until the entire coupled run is finished. During the exchange phase, each D&T process receives the grid point field values (transferred from the source component code with a PRISM_Put call) corresponding to its lists, calculates the regridding weights if it is the first exchange, and applies the weights. The resulting target values are then available for the corresponding target PSMILe process. The data is sent upon request from the respective target process (i.e. when a PRISM_Get is called in the target component code). The calculation of the weights depends on the regridding algorithm chosen by the user in the XML configuration file, which can be different for each coupling field. The OASIS4 D&T therefore acts as a parallel buffer in which the transformations take place.

In OASIS4, the following transformations are available for 2D and 3D coupling fields in the Earth spherical coordinate system for grids that are regular in longitude and latitude, stretched, rotated, or Gaussian reduced (unstructured grids are not supported):

- time accumulation or averaging
- addition or multiplication by a scalar
- gathering/scattering (required when the grid definition includes all masked and non masked points but when the coupling field itself gathers only non masked points)
- 2D nearest-neighbour, Gaussian-weighted, bilinear, bicubic interpolations
- 3D nearest-neighbour, Gaussian-weighted, trilinear interpolations
- 2D conservative remapping
- user-defined regridding (the weights and addresses are pre-defined by the user in an external file; in this case, the PSMILe read these weights and addresses, perform the weights multiplication on the source side and redistribute the results directly from the source to the target)
- global conservation

The parallel global search is implemented for all grids supported and for all regriddings. The 2D algorithms are taken from the Spherical Coordinate Remapping and Interpolation Package (SCRIP) library [Jones1999]. The 3D algorithms are 3D extensions of the 2D SCRIP algorithms and still need to be fully validated.

OASIS4 does not support vector regridding. For an exact regridding, the source vector field has to be projected in a Cartesian coordinate system by the source model and the 3 resulting components have to be provided as separate coupling fields. On the target side, the 3 interpolated components have to be transformed to the local coordinate system after reception.

2.1.6 User community

The current user community of OASIS4 is still quite limited. A first version of OASIS4 was used with pseudo models to interpolate data onto high resolution grids at the Leibniz Institute of Marine Sciences at the University of Kiel (IFM-GEOMAR) in Germany. OASIS4 has also been used for 3D coupling between atmosphere and atmospheric chemistry models at ECMWF, KNMI and Météo-France in the framework of the EU GEMS project. Currently, OASIS4 is used at SMHI for regional ocean-atmosphere coupling applied to the Arctic region, at the Bureau of Meteorology (BoM) in Australia also for regional ocean-atmosphere coupling, and at the Alfred Wegener Institute, (Bremerhaven, Germany) 2D global ocean-atmosphere coupling. OASIS4 is also being assessed for use in HadGEM3 and other coupled systems by the UK Met Office.

2.2 Performances and scalability

Although, OASIS4 is currently being used in several real coupled models, we do not have yet any real measure of performance in these models. We therefore used a simplified case to perform a first test of the PSMILe library and D&T scalability. The test case is a bidirectional exchange of 2-dimensional data between a global “atmospheric” component and an “ocean” component, ranging from 0 to 360 degrees in longitude and latitudes between 70 S and 70 N. The components are partitioned in latitude direction and the search is performed for bilinear interpolation for one exchange field for each direction. In the 2D cases, we use a T255 “atmospheric” grid with 768×385 grid points for component A and an “ocean” grid with 1202×665 grid points for component B. In the 3D cases, the problem is expanded in the vertical dimension towards a full 3-D search and data exchange with 40 levels for component A and 45 levels for component B. These tests are also reported in form of tables in [Redler 2010].

2.2.1 Scalability of the PSMILe initial neighborhood search

For this particular benchmark, the OASIS4 sources have been compiled with the Intel Fortran compiler version 10.1 and the GNU C compiler version 4.1.2, both using default compiler switches without any further optimisation. The code has run on a local PC cluster. Each node of the cluster is equipped with 2.0 GHz 2 times single core AMD Opteron processor 246 and 4 GB of memory per node connected via a 2 Gigabit Myrinet. For the message passing, we used the Message Passing Interface Chameleon Glenn's Messages proprietary communication layer (MPICH-GM) provided by Myricom. When repeating the measurements the differences in time are in the order of a few milliseconds. Therefore, we decided not to include any statistics as this will not change the general picture we are going to discuss.

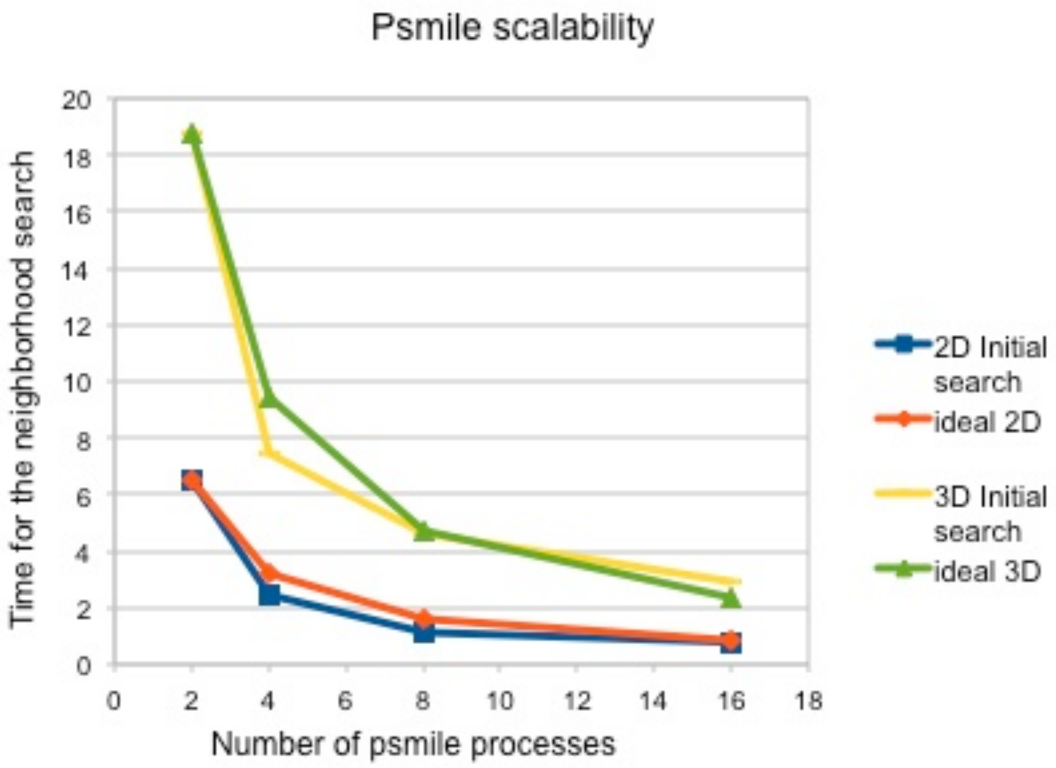

Figure 2 – Time to perform the neighbourhood search as a function of the number of PSMILe processes for the 2D and 3D cases. Ideal curves suppose that the time decreases by a factor X when the number of processes is increased by the same factor X.

Figure 2 presents the time needed to perform the neighbourhood search for the bilinear interpolation for the 2D (in blue) and 3D (in yellow) cases when the number of component processes increase from 2 to 4 to 8 to 16. As the search is done by the source PSMILe but involves some synchronisation between the source and the target components, we observed that this time was almost the same in both components. On Figure 2, we also reported the ideal curve for the 2D (in red) and 3D cases (in green); the ideal curve supposes that the time decreases by a factor X when the number of processes is increased by the factor X. We see that up to 16 processes, the PSMILe shows a very good scalability.

2.2.2 Scalability of the Driver/Transformer (D&T)

Figure 3 shows the time to complete the first ping-pong exchange (in blue) and any of the subsequent ping-pong exchange (averaged over 100 exchanges) (in yellow) as a function of the number of Transformer processes for the 2D case (right) and the 3D case (left).

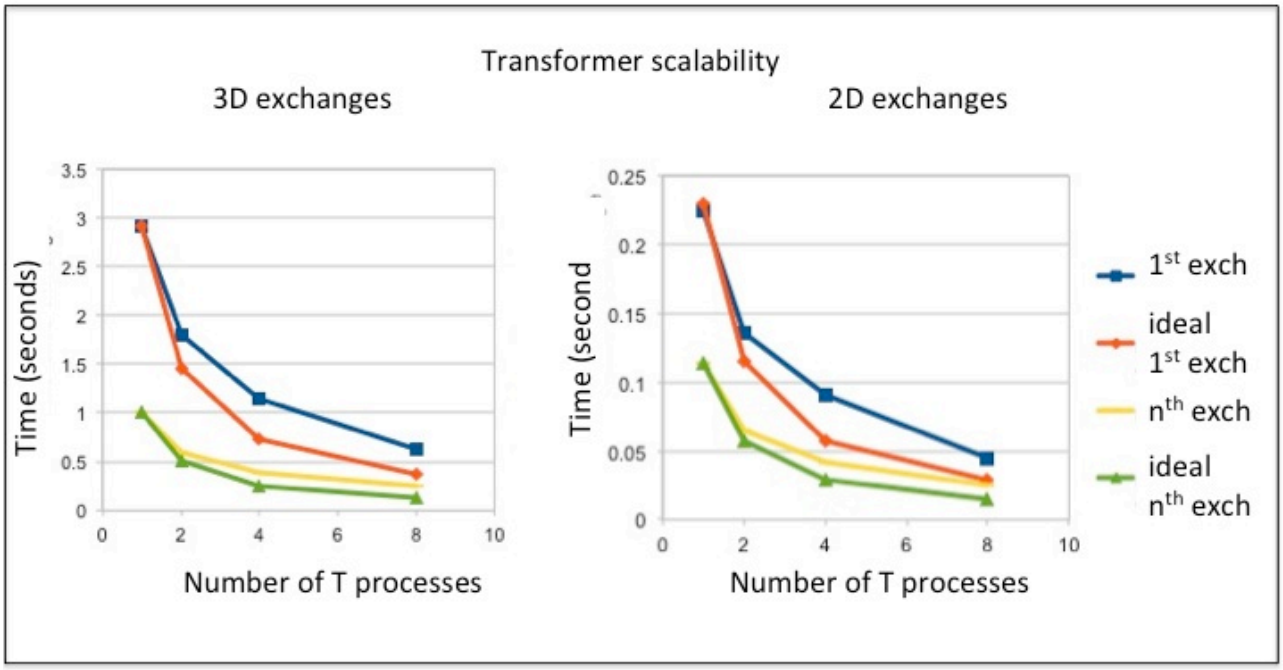

Figure 3 – Time to perform the neighbourhood search as a function of the number of PSMILe processes for the 2D and 3D cases. Ideal curves suppose that the time decreases by a factor X when the number of processes is increased by the same factor X.

In this case also, it was observed that this time is almost the same in the atmosphere and the ocean component. The ideal curves are also shown in red for the first exchange and in green for the subsequent exchanges. Again, we see that up to 16 processes, the D&T (Transformer) shows a good scalability.

These first tests on the PSMILe and the D&T (Transformer) scalability are encouraging and can be used as a proof-of-concept. Of course, they should be completed with additional tests on much greater number of processes before any firm conclusions can be drawn.

2.2.3 Performance of the multi-grid search and parallel regridding

To evaluate the efficiency of the OASIS4 multi-grid algorithm, the test case was adapted to the OASIS3 coupler and additional 2-D coupled runs were realized for different resolutions of components. These additional runs were performed on a Single Core Intel Pentium 4 CPU 3.20 GHz Linux PC with MPICH-1 message passing. OASIS4, OASIS3 and the benchmark sources were compiled with the Portland Group Fortran Compiler 9.0-4 and with the GNU C compiler 4.4.1.

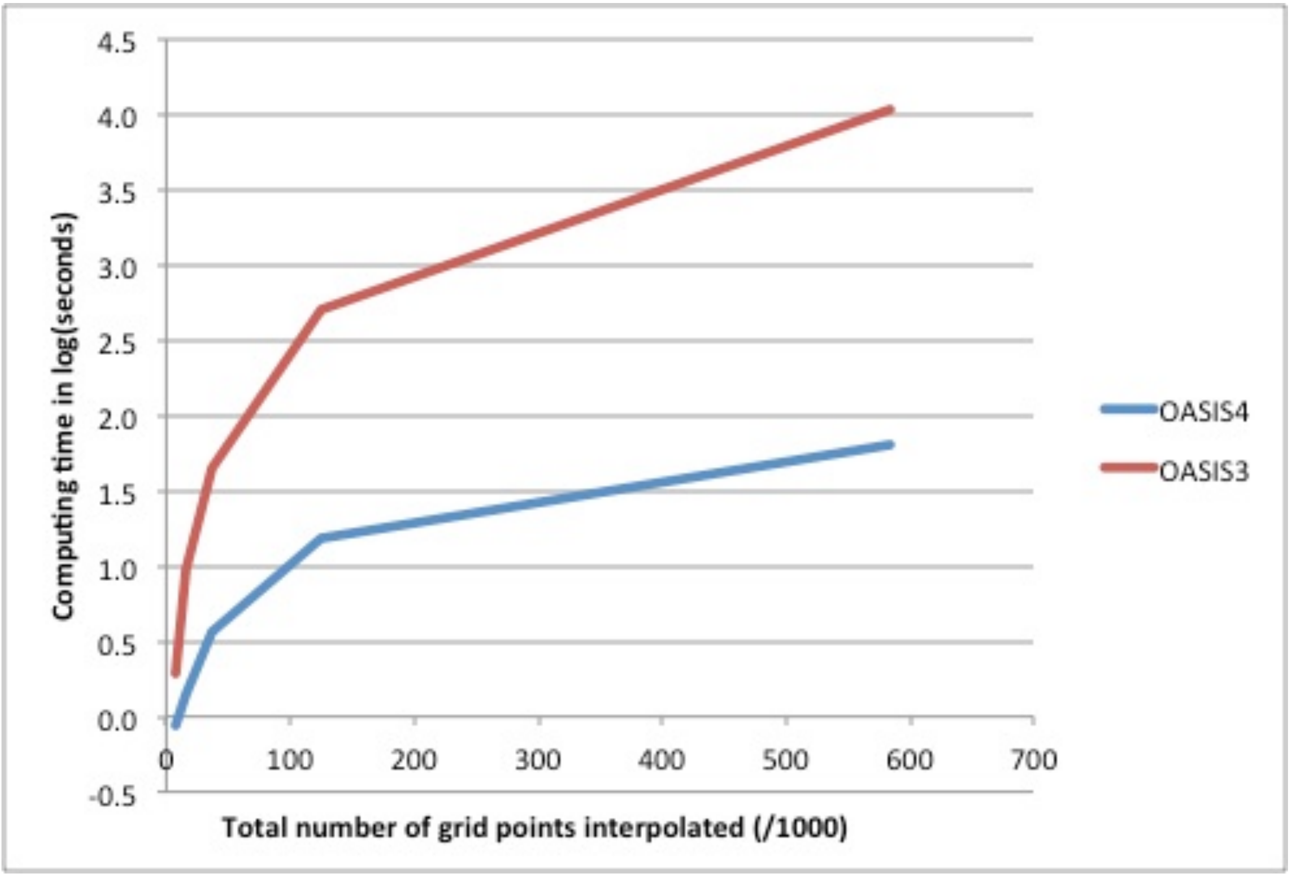

Figure 4 – Time in log(seconds) for the search and the 1st exchange as a function of the total number of points interpolated for the atmosphere and the ocean during the ping-pong exchange for OASIS3 (in red) and for OASIS4 (in blue)

Five runs were realized with OASIS3 and with OASIS4 with a resolution ranging from T21 (2244 grid point) to T255 (295 680 grid points) for the atmosphere, and ranging from 4692 grid points to 288 078 grid points for the ocean. In all cases, the D&T and the components were running with one process each. Figure 4 gives the time for the search and the 1st exchange (again this measure was almost the same for both components). We use these measures to have a comparable basis: with OASIS3, the neighbourhood search and the weight calculation is done during the 1st exchange, where as they are respectively done during the initialisation phase and during the 1st exchange with OASIS4.

Even at relatively low resolution (2244 and 4692 grid points for the atmosphere and the ocean), it is observed that OASIS3 is about two times slower than OASIS4. The difference gets bigger with increasing resolution: in fact, the time required for the neighbourhood search and the first exchange (including the weights calculation) increases with O(N^2) for OASIS3 where as it increases only with O(N*logN) for OASIS4. This clearly demonstrates the benefit of the OASIS4 multi-grid neighbourhood search when compared to the classical OASIS3 search. However, for most current climate models, the grids do not change during the simulation, and so this search is done once at the beginning of the run; its costs becomes therefore less significant compared to the overall model cost when the run gets longer.

2.3 Developments and bug fixes done in IS-ENES

The OASIS4 TRAC ticketing system[2] and wiki pages[3] allow anyone to follow the OASIS4 developments and bug fixes done since the beginning of IS-ENES. The following ones were done during this period (see section 2.4.3 for more detail on the ORCAU, ORCAT, BT42 grids):

1) Developments:
   - Complete rewriting of the parallel global search for Gaussian Reduced grids: see closed tickets #52, #104;
   - Support of grids with multiple blocks per partition, i.e. one process treats two or more disconnected regions of points – e.g. used in ECHAM5 parallelization: see closed tickets #59, #98; a multiblock partition is illustrated at Figure 5 B; (all partitions supported by OASIS4 are described in section 5.3.4 of OASIS4 User Guide[4])

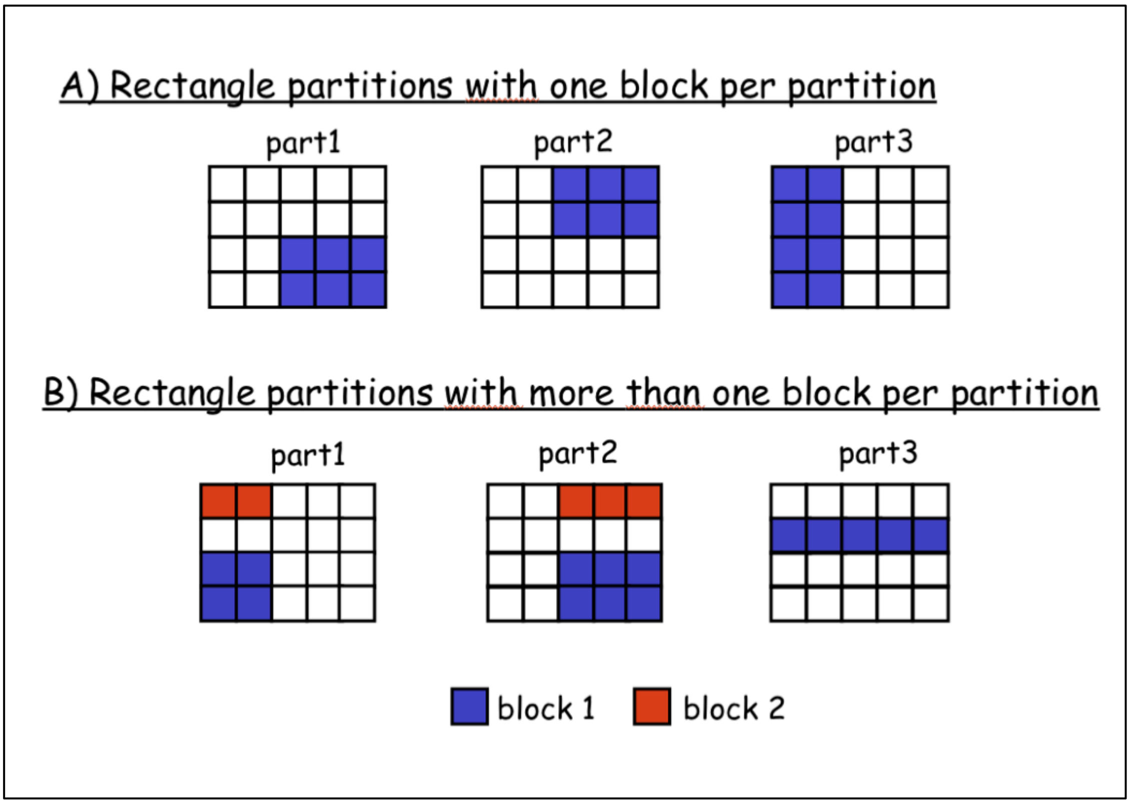
A) Rectangle partitions with one block per partition
part1
part2
part3
B) Rectangle partitions with more than one block per partition
part1
part2
part3
block 1
block 2

Figure 5 – A) mono and B) multi block partitions supported logically rectangular grids

   - Validation, testing and bug-fixing of the global search for the 2D conservative remapping: see closed ticket #11);
   - Special treatment for tripolar grids, such as ORCA or MPI-OM: see closed ticket #80, #111, and #119;
   - Implementation of the possibility to use user-defined sets of weights-and-addresses for the regridding[5]: see closed ticket #16;
   - Development of a Graphical User Interface[6] for the realization of the XML configuration files;
   - Simplification of the XML configuration files and optimisation of the XML reading[7]: see closed ticket #50;
   - Improvement of the performance of the exchanges between a component PSMILe and the

[2] https://oasistrac.cerfacs.fr/report
[3] https://oasistrac.cerfacs.fr/wiki/OASIS4Development
[4] http://www.cerfacs.fr/oa4web/oasis4/OASIS4_User_Guide.pdf
[5] See http://pantar.cerfacs.fr/globc/publication/technicalreport/2011/DG_User-Defined-Interpol.pdf
[6] See http://pantar.cerfacs.fr/globc/publication/technicalreport/2010/OASIS4-GUI_userguide.pdf
[7] See http://pantar.cerfacs.fr/globc/publication/technicalreport/2010/Optimisation_de_la_lecture_des_fichiers_XML.pdf

D&T : see closed ticket #57;

- Modifications to ensure that the behaviour of the OASIS4 coupled system in case of error is clean: see closed ticket #91;
- New simpler directory structure for OASIS4 sources.
- A toy model reproducing the coupling exchanges of the full ARPEGE-NEMO coupled system with realistic grids and partitions has been developed. It was used in particular to validate the different partitions used for Gaussian Reduced grids: with complete latitudinal bands (see Figure 6 C), with partial latitudinal bands (see Figure 6 D), with non consecutive partial latitudinal bands (see Figure 7). See also closed ticket #94.

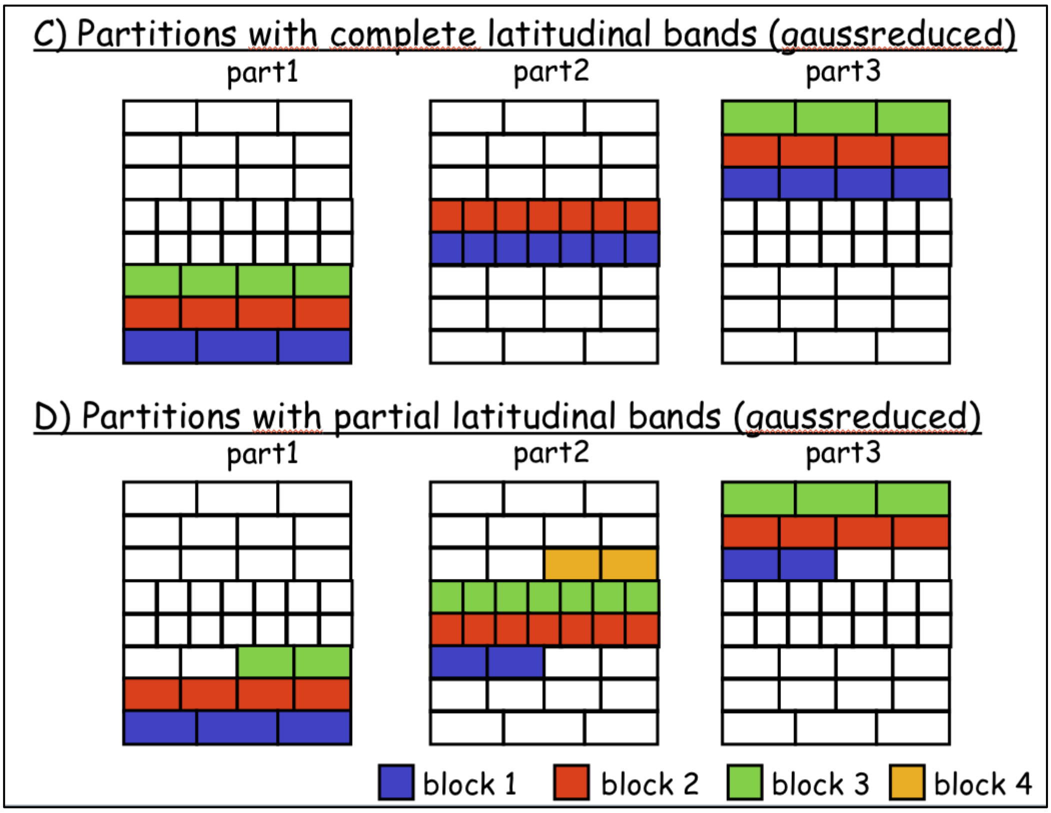
C) Partitions with complete latitudinal bands (gaussreduced)
part1
part2
part3
D) Partitions with partial latitudinal bands (gaussreduced)
part1
part2
part3
🟦 block 1
🟥 block 2
🟩 block 3
🟧 block 4

Figure 6 – Partitions with C) complete or D) partial latitudinal bands for Gaussian reduced grids

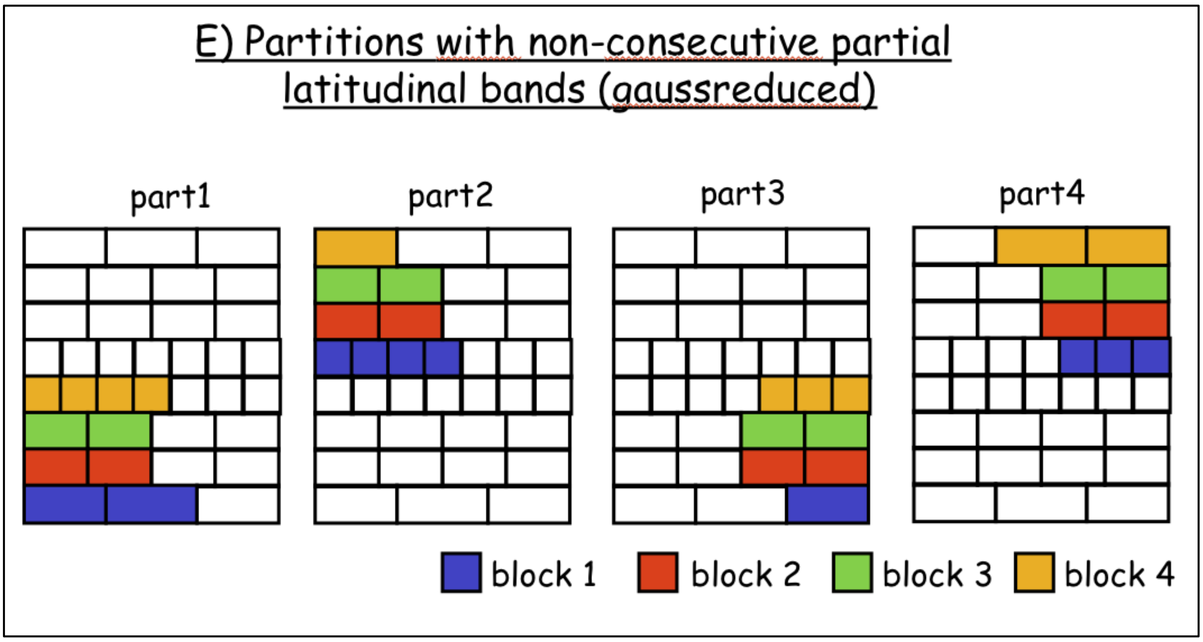
E) Partitions with non-consecutive partial latitudinal bands (gaussreduced)
part1
part2
part3
part4
block 1
block 2
block 3
block 4

Figure 7 – Partitions with non-consecutive partial latitudinal bands for Gaussian reduced grids

2) Bug fixes:
   - Bug fix to compile with NAG/5.2.668 compiler: see closed ticket #54;
   - Bug fix for bicubic interpolation related to periodicity of the grid: see closed ticket #67;
   - Bug fix for arbitrary coordinate ranges: see closed ticket #68;
   - Clarification of the use of the SMIOC periodic attribute in the XML configuration files: see closed ticket #69;
   - Bug fix for the problem observed at 180° in the ORCAT-BT42 interpolation: see closed ticket #76;
   - Bug fix for the conflict between the regridding parallel global search and the nearest-neighbour extra search: see closed ticket #103;
   - Bug fix for the bicubic interpolation for Gaussian Reduced grids: see closed ticket #106.
   - Bug fix for the ORCAU-BT42 bicubic sixteen interpolation: see closed ticket #110.

Finally, it is worth mentioning that, as part of IS-ENES WP4/NA3 Task 2, OASIS4 User Guide has been updated and is now available on line[3] on the new OASIS4 web site[8] in the IS-ENES portal.

Interactions between the developers occurred mainly through daily e-mail exchanges. Three two-day developers meetings were also organised to allow closer exchanges, one in May 2009 in Toulouse[9], one in October 2009 in Hamburg[10] and one in September 2010 in Hamburg[11].

2.4 Validation: toy models and regridding quality

This section describes the tests done to validate the general use of OASIS4 to set up a coupled system (see the toy model "toyoa4" in section 2.4.2 below) and in particular the user-defined, bilinear, bicubic and conservative regridding for the grids described in 2.4.3 (regular longitude-latitude, logically rectangular stretched, Gaussian reduced), and for the following types of parallel partitioning:

- Rectangle partitions with one or more blocks per partitions for logically rectangular grids (see Figure 5)
- Partitions with complete or partial latitudinal bands for Gaussian reduced grids (see Figure

[8] https://verc.enes.org/models/software-tools/oasis
[9] See minutes at http://pantar.cerfacs.fr/globc/publication/technicalreport/2010/200905_DevMeeting.pdf
[10] See minutes at http://pantar.cerfacs.fr/globc/publication/technicalreport/2010/200910_DevMeeting.pdf
[11] See minutes at https://oasistrac.cerfacs.fr/wiki/DeveloperMeetingMPI2010

6)
- Partitions with non consecutive partial latitudinal bands for Gaussian reduced grids (see Figure 7)

By extension, although they were not validated, partitions with partial latitudinal bands or even non-consecutive blocks should be supported for regular longitude-latitude grids if declared as Gaussian reduced.

The problems encountered that still need to be solved for a fully successful validation are detailed in section 2.5 below.

2.4.1 Use of Buildbot for automation of validation tests

OASIS4 functions have been continuously tested and validated during the development phase, by running toy models that use specific OASIS4 functions (see 2.4.2) and by performing offline regridding quality tests (see 2.4.3). To facilitate this work, Buildbot, which is a software written in Python to automate compile and test cycles required in software project, is used since the last 6 months on different computers and has proven to be a very powerful tool to continuously validate the developments done in the code.

The first step was to define a reference state of some test examples once they were entirely running successfully. Then, at each modification of the OASIS4 sources, we now use Buildbot to test that the results obtained with the new version are the same than for the reference state or possibly better. The different toy models are tested on a Linux PC “tioman”, an IBM Nehalem cluster “octopus” and an HP AMD cluster “corail” at CERFACS, the NEC SX8 “tori” and SX9 “yuki” of Meteo-France, the BULL Novascale 3045 “platine” at CCRT (Bruyères-le-Châtel, France) and the Linux cluster “tornado” at DKRZ (Hamburg, Germany). The offline regridding quality tests run only on the Linux PC “tioman” at CERFACS.

The compilation of OASIS4 is performed every morning on each computing platform described above thanks to a cron defined on a local Linux platform. The different toys are tested at night with Buildbot (installed also on the local Linux platform) if a modification in the sources of OASIS4 is detected via SVN since the last build. The configuration file of Buildbot contains calls to scripts we wrote corresponding to different tasks to be performed on the toys. The scripts are called sequentially for a particular toy but the different tests are launched and run at the same time for the different toys. All tests are first run in monoprocessor mode (i.e. with one process for each model and one process for OASIS4 D&T) and then in parallel (i.e. with more than one process for each model) and some files are created with the results. Then a script verifies that the test ran until completion, that the statistics written in OASIS4 log files are good (comparing the files just created to the ones of the reference state), and that the exchanged fields are the same than the ones of the reference state. In parallel, the exchanged fields are also compared to the monoprocessor case that just ran.

The results of Buildbot are graphical (see http://memphis.cerfacs.fr:8011/waterfall and http://memphis.cerfacs.fr:8013/waterfall[12]) and very easy to analyse, allowing to make many tests each night. The graphical results are illustrated on Figure 8. On the web pages above, the different toys are listed at the top of the page (with one column for each toy on each test platform). The green and red boxes correspond to the different tasks. The box is green if the task was validated; it is red otherwise. This graphical validation is possible because the "exit" command of the shell of the scripts returns a code interpreted by the Buildbot: if the code is 0 the task is validated, else the task is not validated.

[12] Specific access right is needed for these pages; please ask author of this report

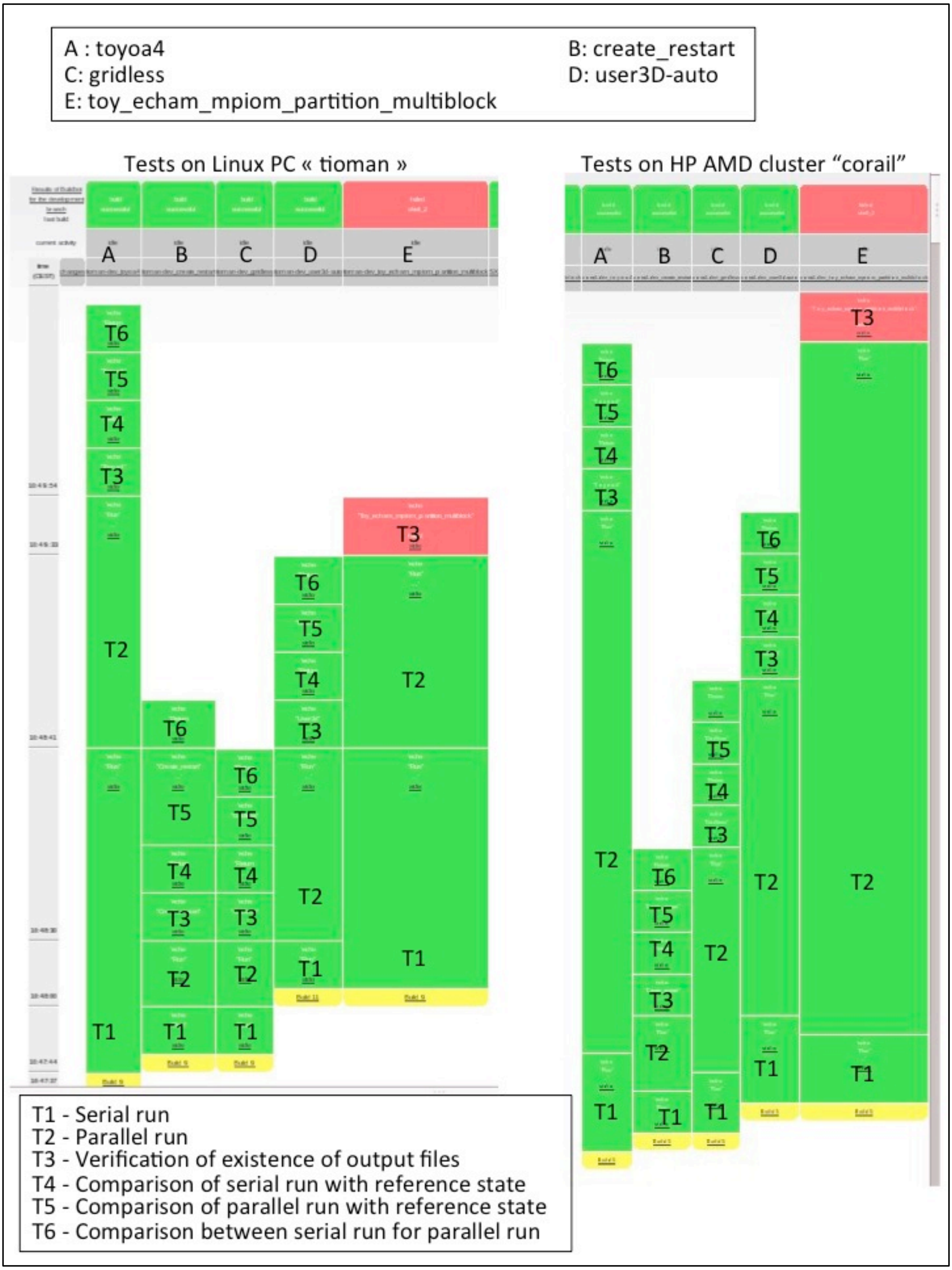
A : toyoa4
B: create_restart
C: gridless
D: user3D-auto
E: toy_echam_mpiom_partition_multiblock
Tests on Linux PC « tioman »
Tests on HP AMD cluster "corail"
A
B
C
D
E
T1
T2
T3
T4
T5
T6
T1 - Serial run
T2 - Parallel run
T3 - Verification of existence of output files
T4 - Comparison of serial run with reference state
T5 - Comparison of parallel run with reference state
T6 - Comparison between serial run for parallel run

Figure 8 – Graphical results as they appear on the buildbot web page showing the validation (green box) or failure (red box) of different tasks T1 to T6 for different test cases A to E on Linux PC "tioman" (left) and HP AMD cluster "corail" (right). Here it can immediately be concluded that task T3 – Verification of existence of output files failed for the toy_echam_mpiom_partition_multiblock test case on both platforms.

2.4.2 Toy models used for validation

The following toy models were routinely run and validated with Buildbot:

1) Toyoa4

   The Toyoa4 toy model is described in all details in section 7.3 of the OASIS4 User Guide. Toyoa4 reproduces a coupled system composed of 3 components with realistic grids: two components, “atmoa4” and “lanoa4” work on a T31 Gaussian Reduced grid; the third model “oceoa4” uses a real ocean model stretched and rotated grid with spherical polar coordinates of 182x149 grid points. In total 6 coupling fields (which are simply equal to a constant value, different for each field) are exchanged between the different components and different regriddings and transformations are applied. Two configurations are tested and results of the regriddings are available on OASIS4 wiki: in the first one all components run onto one process each[13]; in the second one, “atmoa4”, “lanoa4” and “oceoa4” run respectively onto 3, 2, and 2 processes and the partitioning is done along the second index (j) direction for all components[14]. For each test, it is checked that it runs until completion and that the regridding of the coupling fields do not give abnormal results.

2) Toy_echam_mpiom_partition_multiblock

   This toy example tests the multiblock functionality and uses the grids of the ECHAM5/MPI-OM coupled system. The toy atmospheric component runs on the ECHAM5 T127 regular (384x192) grid and is partitioned on 4 processes with two blocks on each process along the first index (i) direction. The ocean component runs on MPI-OM tripolar (800x402) grid and is not parallel. The regridding used is bilinear. For each test, it is checked that it runs until completion and that the regridding of the coupling fields do not give abnormal results[15].

3) User3d-auto:

   This toy example tests the user-defined regridding (see 2.1.5 above and section 4.3.3 of OASIS4 User Guide). The source model defines a grid of (180x91) grid points and the target model a grid of (360x181) grid points. The pre-defined set of weigths and addresses associates 10 source grid points to 7 target grid points with different weights. For each test, it is checked that it runs until completion and that the 7 regridded values are the correct ones.

4) Gridless:

   This toy example tests an exchange of a coupling field on a non-geographical grid (“gridless” grid, see section 5.3.7 of OASIS4 User Guide) of 1000 points on both the source and target side. The source and the target are respectively partitioned onto 3 and 2 processes and for each test, it is checked that the repartitioning of the field between the source and the target gives correct values on the target.

5) Create_restart:

   This is an example to test the explicit creation of OASIS4 coupling restart files with routine prism_put_restart. The standalone application, running on a (64x128) grid, creates 19 restart fields in the file dummy.nc. On each grid point, the field value is equal to its index in the first dimension. For each tests, it is checked that it runs until completion and that the restart fields created have correct values; the values of 4 restart fields is illustrated on the OASIS4 wiki[16].

[13] http://www.cerfacs.fr/oa4web/results_examples_oa4_devr3211/toyoa4_np1_mono/index_12figures.html

[14] http://www.cerfacs.fr/oa4web/results_examples_oa4_devr3211/toyoa4_np1_para/index_12figures.html

[15] http://www.cerfacs.fr/oa4web/results_examples_oa4_devr3211/toy_echam_mpiom_partition_multiblock_np1_para/index_4figures.html

[16] http://www.cerfacs.fr/oa4web/results_examples_oa4_devr3211/create_restart_np0_mono/index_4figures.html

2.4.3 Offline regridding quality tests

The quality of bilinear, bicubic and 2D conservative OASIS4 regriddings are routinely checked in monoprocessor and parallel modes. A set-up[17] has been developed to easily test different regriddings for different grids in different parallel configurations. The coupling fields are defined with an analytical function in the source model, appl-atm.F90. The coupling fields are sent, interpolated by OASIS4 and received by the target, appl-ocn.F90. The regridding error is then calculated on each target grid point as the difference between the interpolated field and the value of the analytical function at the same grid point (divided by the analytical value and multiplied by 100 to have the error in %).

The following grids are tested:

- ORCA2T, ORCA2U, ORCA2V: NEMO ocean model tripolar stretched T, U or V grids (logically rectangular), ~2 degree resolution, 182x149x1 grid points
- BT42: ARPEGE atmospheric model Gaussian Reduced grid, 6232x1 grid points
- LMDZ: LMDz atmospheric model regular longitude-latitude grid, 96x72x1 grid points
- ALADIN: Aladin regional atmospheric model logically rectangular grid with 117x69x1 grid points
- MED1/2T, MED1/2U, MED1/2V: Mediterranean regional ocean model logically rectangular grid with 188x88x1 grid points

Different combinations of these grids for different regriddings are tested. All results are available on OASIS4 wiki for the monoprocessor case at http://www.cerfacs.fr/oa4web/projet_cicle_trunkr3248/RESULTS_BASSE_RESOLUTION/projet_cicle.html and for the parallel mode at http://www.cerfacs.fr/oa4web/projet_cicle_trunkr3248/RESULTS_BASSE_RESOLUTION_PARALLELE3PROCS/projet_cicle.html . In parallel different partitionings are used and the one used for each test is explicitly named in the first column of the result table:

- PARTI X 1: means that the grid is split into X parts along the first (i) dimension
- PARTJ 1 Y: means that the grid is split into Y parts along the second (j) dimension
- PARTIJ X Y: means that the grid is split into X parts along the first (i) dimension and Y parts along the second (j) dimension

For each regridding, the results are accessible on a particular web page containing four frames: the top left and right ones show respectively the field before and after regridding, and the bottom ones show the regridding error with two different colour scales.

2.5 Remaining problems in OASIS4_1beta

As stated above, different problems were encountered during OASIS4 development and validation phases. Many of them were solved and the details of the bug fixes applied are described in the closed tickets on OASIS4 TRAC system at https://oasistrac.cerfacs.fr/report. The most important bug fixes are also listed at the beginning of section 2.3 above. However, different issues and problems still remain. As the time was too short to try to solve them all for this deliverable and as the effort devoted to these developments is already higher than expected for the whole project (see section 3.0 below), it was decided to release the current sources tagging them with tag "OASIS4_1beta" to make clear that this is not yet a stable version of OASIS4.

The remaining problems and issues are listed here.

1. The target points that fall outside the source grid or in a hole of the source grid do not receive any value at all (see details and the discussion in ticket #29). This causes in particular different results between the monoprocessor mode (i.e. with one process for each model and one process for OASIS4 D&T) and the parallel mode (i.e. with more than one process for each model) for the conservative remapping for the target cells that overlap the border of a source partition and that have some corners falling into holes of the source grid. In monoprocessor mode, the whole target cell surface is considered but in parallel, only the surface intersecting the process domain with no holes is used for the remapping

[17] The sources are available at https://oasistrac.cerfacs.fr/browser/trunk/prism/dev_ex/test_interpolations/src

normalisation (ticket #29, note dated 04/08/11).

2. Very small differences of $O(10^{-14}$-$10^{-15})$ between the monoprocessor and parallel modes appear near the border of the source partitions; this difference could be caused by a different order in the summation of the line integral but this still has to be verified (see details in ticket #64)
3. For the conservative remapping, few points get different values in monoprocessor and parallel modes along the borders of the source partitions (see tickets #120, #122, #123) and at the North (see tickets #121 and #124): these problems have not been investigated yet.
4. Some problems occur for the conservative remapping but are probably not linked to OASIS4 itself but to the use of the SCRIP library (tickets #4 and #114)
5. A deadlock is observed for the conservative remapping of the multiblock partitioning of ECHAM (ticket #100)
6. For coupling fields defined on multiblock partitions, the I/O and therefore in particular the reading and the writing of the restart files are not supported (see ticket #60)
7. The cell area calculated by the SCRIP library, i.e. defined by segments linear in longitude and latitude going from one corner to the other, is not the true area. Normalisation by the true area (i.e. the one considered by the model) is needed to ensure conservation (ticket #7)
8. Running two components sequentially within one application, although implemented, produces a deadlock (see ticket #96)
9. Some attempts to implement OASIS4 into real coupled systems as part of the OASIS Dedicated User Support offered in the framework of IS-ENES NA3/WP4 also revealed specific problems[18]:
   - In the COSMO-CML coupled system (ETH-Zurich, Switzerland), a mysterious slow down in COSMO-CML coupling was observed when the number of processes of CML was increased (even if they are not implied in the coupling exchanges);
   - In the NEMO-WRF coupled system (IPSL, Paris, France), failure during the parallel nearest neighbour interpolation weight calculation was observed for more than 128 cores.

Additionally, the following developments should be done for a fully operational coupler:

10. For the conservative remapping, a target cell that intersects only masked source cells does not receive any value; an extra nearest-neighbour search (see 2.1.4.i) should be implemented as an option (see details in ticket #32)
11. The extra nearest-neighbour search needs also to be implemented for the nearest-neighbour interpolation as an option when the “true” nearest neighbours are all masked (see ticket #81).
12. Forced global conservation should be implemented to ensure conservation even when the source and target coastlines do not match; this can be done by distributing over the non masked target grid points the difference between the integral of the field on the source non masked source grid points and the integral of the regridded field on the non masked target grid points (see ticket #43)

3. Discussion and next steps

Since the beginning of the IS-ENES project (~27 months), two persons (one engineer at DKRZ and one CNRS agent at CERFACS) have worked full time and two additional persons (one engineer at CERFACS and one at MPI-M) have worked part time on OASIS4 development and validation. The effort devoted to this task is therefore already much more than the 57 pm associated to IS-ENES

[18] See details in 2010 report available on IS-ENES internal web site at https://is.enes.org/eu-internal/exchange-platform-2/na3/oasis/copy_of_dedicated-user-support/2010_OUS_report_2.pdf

WP7/JRA1 Task 2. However, given the remaining problems described in the last section, we are still not able to deliver a stable version of OASIS4 parallel neighbourhood search library.

It must be concluded at this point that our first objective, which was to implement a fully parallel regridding library, is out of reach given the current resources. Furthermore, our analysis is that the current OASIS4 code base is not a stable basis to build on as it has reached a point where it is too complex to evolve easily to answer current and future climate modeling coupling needs. In particular, the support of unstructured grids was not included in the original design and it now would be very difficult to add it in the current code. However, this conclusion, although unsatisfactory, does not prevent the IS-ENES partners from performing coupled climate modelling simulations at resolutions used today and potentially at even higher resolutions. In fact, the OASIS3 coupler with its field-per-field parallelisation still gives satisfactory results for the resolution currently used in operational coupled climate models. In particular, OASIS3 is currently used in 5 of the 7 European ESMs participating to CMIP5 without causing any significant overhead. OASIS3 has even been used recently in few higher-resolution coupled simulations without introducing significant overhead in the simulation elapsed time. For example, an overhead of less than 3% was measured in a recent simulation of EC-Earth using O(1000) cores into which the atmospheric component IFS T799 (~25 km, 843 000 pts) and 62 vertical levels was coupled to the NEMO ocean model using the ORCA0.25 configuration (1.5 Mpts) and 45 depth levels (see Appendix A for details). It can therefore be concluded that OASIS3, thanks to its field-per-field parallelisation, still addresses the needs of current coupled climate models, which was the objective of the current deliverable.

It is however expected that for more than O(1000) cores, which is already close to the minimum limit of cores required to run on PRACE machines, the limited parallelism of OASIS3 will soon become a bottleneck in the coupled simulation and the next steps in IS-ENES have to address this issue. It is important to realize that the fully parallel calculation of the regridding weights and addresses, implemented in OASIS4 with only limited success as detailed in this report, is certainly not mandatory for our climate models in the short and mid term (~5 years, next IPCC report). In fact, this functionality will actually only be needed only when the component models will run on adaptive grids (which possibly change at each timestep) or when the sequential offline calculation of the weights and addresses will no longer be possible, because the memory of one core will not be sufficient to hold the entire global grid.

Therefore, it is proposed here to focus the remaining OASIS4 development efforts in IS-ENES on the user-defined regridding functionality (see above and also section 4.3.3 of OASIS4 User Guide), which bypasses the parallel neighbourhood search. To use this functionality, the user has to provide, in a separate NetCDF file, the links associating specific source grid points with specific target grid points; for each link, the index of the source point and the index of the target point (in the total source grid dimension and target grid dimension respectively), and the weight associated to that link. The OASIS4 PSMILe library reads these indices and weights and automatically defines on each side a non-geographical (gridless) grid with one point for each link. The multiplication of the source field values by the appropriate weights is done on the source side and parallel redistribution of the results is done directly between the source and the target processes. The user-defined regridding is implemented and validated with simple toy models; it now has to be validated for real grids and in real coupled ESMs. This functionality will therefore remove the predicted bottleneck of the current OASIS3 version of the coupler and should allow our current and next climate models to run efficiently on massively parallel platforms on more than 1000 cores. It also provides an important level of traceability for scientists who need to be able to demonstrate the integrity of their models.

4. Conclusions

This document describes the work done on the OASIS4 parallel coupler since the beginning of the IS-ENES project. A general description and results of scalability tests performed on the current OASIS4 version, OASIS4_1beta, are first reported. All developments, bug fixes and validation tests done in IS-ENES are then detailed and known remaining problems are listed.

The discussion in section 3 highlights the fact that although the efforts devoted since the beginning of the project have already exceeded the total planned for the whole project, we are still not able to deliver a stable version of OASIS4 parallel neighbourhood search library. Therefore, we have to conclude that our first objective, which was to implement a fully parallel regridding library on the base of the OASIS4 sources, is out of reach given the current resources.

We also detail in the discussion how the OASIS3 version of the coupler still addresses the needs of current coupled climate models coupler thanks to its field-per-field parallelisation. This means that the negative conclusion reached for the global parallel interpolation library of OASIS4 is not a blocking issue for current coupled climate modelling simulations.

However, since it is expected that for more than O(1000) cores, the limited parallelism of OASIS3 will become a bottleneck in the coupled simulation, we propose concrete steps to address this issue. As the OASIS4 fully parallel calculation of the regridding weights and addresses is not critical for our climate models in the short and mid term, we propose here to focus the remaining OASIS4 development efforts in IS-ENES on its user-defined regridding functionality, which bypasses the parallel neighbourhood search. To use this functionality, the user has to provide the regridding weights and addresses, which are then read and used by the OASIS4 communication library that performs a parallel redistribution of the source coupling data directly between the source and the target processes. This functionality therefore removes the foreseen bottleneck of the OASIS3 coupler and should allow our current and next climate models to run efficiently on massively parallel platforms.

A future version of OASIS4 will be released before the end of IS-ENES including a fully tested and debugged user-defined regridding functionality but no further development of its parallel neighbourhood search library will be pursued. In the mean time, we recommend that climate modellers keep on using OASIS3 version that still gives satisfactory results for most of current climate coupled models.

References

1. [Valcke2006a] S. Valcke, E. Guilyardi, C. Larsson, 2006. PRISM and ENES: A European approach to Earth system modelling. *Concurrency Computat.: Pract. Exper.,* 18(2), pp. 231-245.
2. [Redler2010] R. Redler and S. Valcke, 2010. OASIS4, A Coupling Software for Next Generation Earth System Modelling. *Geosci. Model Dev.,* 3, 87-104.
3. [Balaji2001] Balaji, 2001: Parallel Numerical Kernels for Climate Models, ECMWF TeraComputing Workshop 2001, World Scientific Press, Reading.
4. [Jones1999] Jones, P.: Conservative remapping: First- and second-order conservative remapping, Monthly Weather Review, 127, 2204–2210, 1999.
5. [Gropp1998] Gropp, W., S. Huss-Lederman, A. Lumsdaine, E. Lusk, B. Nitzberg, W. Saphir, and M. Snir, 1998: MPI -- The Complete Reference, Vol. 2 The MPI Extensions, MIT Press.
6. Snir1998] Snir, M., S. Otto, S. Huss-Lederman, D. Walker and J. Dongarra, 1998: MPI – The Complete Reference, Vol. 1 The MPI Core, MIT Press

APPENDIX A – OASIS3

OASIS3

OASIS3[19] is the direct evolution of the OASIS coupler developed since 1991 at CERFACS and widely used around the world for current coupled climate model. Like OASIS4, the sources of OASIS3 form, after compilation, a separate executable that performs the regridding for the coupling fields and a communication library that needs to be linked to the component models. This appendix provides a general description of OASIS3 and a brief presentation of some recent performance measurements.

Coupling configuration

The coupling configuration is very similar in OASIS3 and in OASIS4. At the beginning of the run, the OASIS3 separate executable reads the coupling configuration defined by the user before the run and distributes the corresponding information to the different component model PSMILes. For OASIS3, this user-defined configuration, which contains all coupling options for a particular coupled run is provided in a text file. During the run, the OASIS3 executable and the component model PSMILes perform appropriate exchanges based on this configuration.

Process management

The process management in OASIS3 is identical to the process management in OASIS4. MPI1 and MPI2 modes are available for starting the executable components of the coupled application, which are necessarily integrated from the beginning to the end of the run. The component models remain separate executables with main characteristics untouched with respect to the uncoupled mode.

Communication: the OASIS3 PSMILe library

OASIS3 supports partially parallel communication in the sense that each process of a parallel model can send or receive its local part of the field. With the OASIS3 PSMILe, the different local parts of the field are sent to the OASIS3 central executable that gathers the whole coupling field, transforms or regrids it, and redistribute it to the target component model processes. With the last version of OASIS3, it is possible to run the central executable on more than one process, each process treating a subset of the complete coupling fields. This field-per-field parallelisation allows a certain degree of optimisation as detailed below in the performance section.

Coupling field transformation and regridding in OASIS3

In OASIS3, all steps of the regridding (neighbourhood search, weight calculation) are done by the central executable at the beginning of the run considering the whole source grid. The same 2D transformations than in OASIS4 are available for grids that are regular in longitude and latitude, stretched, rotated, Gaussian reduced, but also unstructured:

- time accumulation or averaging
- addition or multiplication by a scalar
- nearest-neighbour, Gaussian-weighted, bilinear, bicubic 2D interpolations
- 2D conservative remapping

[19] More details are available in OASIS3 User Guide at http://www.cerfacs.fr/oa4web/oasis3/oasis3_UserGuide.pdf

- user-defined regridding

In addition, the following transformations are available in OASIS3:

- linear combination with other coupling fields
- correction with external data read from a file
- global conservation
- creation of subgrid scale variability (when regridding from low to high resolution)

As in OASIS4, the interpolations and conservative remapping are taken from the SCRIP library. OASIS3 also supports regridding of vector fields with the projection of the two vector components in a Cartesian coordinate system, regridding of the resulting 3 Cartesian components, and projection back in the spherical coordinate system. OASIS3 can also be used in the interpolator-only mode to transform and regrid fields contained in files without running any model.

User community

OASIS3 user community reaches today about 30 climate modelling groups. OASIS success up to now can be explained by its great flexibility, the active support offered by the development team to the users, and the great care taken to constantly integrate the community developments in the official version.

OASIS3 in particular is used today by many different climate modelling groups in Europe, Australia, Asia and North America among which Météo-France and the Institut Pierre-Simon Laplace (IPSL) in France, the European Centre for Medium range Weather Forecasts (ECMWF), the Max-Planck Institute for Meteorology (MPI-M) in Germany, the Met Office and the National Centre for Atmospheric Science (NCAS) in the UK, the ``Koninklijk Nederlands Meteorologisch Instituut'' (KNMI) in the Netherlands, the Swedish Meteorological and Hydrological Institute (SMHI) in Sweden, the ``Istituto Nazionale di Geofisica e Vulcanologia'' (INGV) and the``Ente Nazionale per le Nuove tecnologie, l'Energia el Ambiente'' (ENEA) in Italy, the Bureau of Meteorology (BoM) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO) in Australia, the Université du Québec à Montréal and the Service Météorologique du Canada, the Hawaii University and the Oregon State University in the USA, the Institute of Atmospheric Physics from the Chinese Academy of Sciences and Meteorological National Center in China.

OASIS3 performances

The OASIS3 coupler is certainly limited in parallelism and will eventually become a bottleneck in the simulation on massively-parallel platforms. However, thanks to its parallelisation on a field-per-field basis, OASIS3 has been used recently in few high-resolution coupled simulations without introducing significant overhead in the simulation elapsed time. The following coupled models were run with OASIS3:

a) In the high-resolution version of the Hadley Centre coupled model, OASIS3 is used to couple the atmospheric Unified Model (UM) with a horizontal resolution of 432 x 325 grid points (140 000 pts) and 85 vertical levels to the ocean NEMO including the CICE sea ice at a horizontal resolution of ¼ degree (ORCA0.25 configuration, 1.5 Mpts) and 75 depth levels. The coupling exchanges are performed every 3 hours and the coupled model is run on an IBM power6 192 cpus for the UM, 88 cpus for NEMO, and 8 cpus OASIS3. A coupling overhead (in elapse time) of less than 2% was observed.

b) OASIS3 is also used in the high-resolution version of IPSLESM, coupling the LMDz atmospheric model with 589 000 pts horizontally (~1/3 degree) and 39 vertical levels to the NEMO ocean model in the ORCA0.25 configuration (1.5 Mpts) and 75 depth levels on the CINES SGI ALTIX ICE. The coupling exchanges are performed every 2 hours. The coupled system uses up to 2191 cpus, with 2048 for LMDz, 120 for NEMO, and 23 for OASIS3. No detailed performance numbers are available but no significant bottleneck was observed in this configuration.

c) Recently, the resolution of EC-Earth was increased for the atmospheric model IFS to T799 (~25 km, 843 000 pts) and 62 vertical levels and to the ORCA0.25 configuration (1.5 Mpts) and 45 depth levels for NEMO ocean model. This was run on the Ekman cluster (1268 nodes of 2 quadripro AMD Opteron, i.e. a total of 10144 cores) with different numbers of cores for each component and OASIS3. Different combinations were tested and the following table illustrates the benefit of the OASIS3 pseudo-parallelisation:

IFS-NEMO-OASIS nb of cores	512-128-1	512-128-10	800-256-1	800-256-10
1-IFS standalone	41.	41.	29.9	29.9
2-EC-Earth3	45.7	42.3	33.2	30.3
2.1-IFS component	41.8	n/a	32.7	n/a
2.2-NEMO component	38.5	n/a	24,6	n/a
2.3-OASIS	5.5	n/a	6	n/a
Coupling overhead (2-1)	4.7 (13.4%)	1.3 (3%)	3.3 (11%)	0.4 (1.3%)

Table 1: 2 hour long simulation response time (in seconds) for the different components and for EC-Earth3 coupled model. The coupling extra cost is calculated as the difference between EC-Earth and IFS standalone elapse time.

In this configuration, IFS and NEMO run concurrently and not sequentially. We can observe here that OASIS3 elapse time is non negligible when it runs in mono-processor mode (respectively 5.5 seconds and 6 seconds for the 512-128-1 and the 800-256-1 configurations). In this case, the coupling induces significant overhead in elapse time with respect to the IFS standalone run (respectively 13.4% and 11%); this is true even if OASIS3 interpolates the fields when the fastest component waits for the slowest as OASIS3 cost itself is larger than the component imbalance.

But when the parallelism of OASIS3 increases (going from 1 to 10 processes), OASIS3 elapse time decreases and its cost can almost be “hidden” in the component imbalance. Even if we do not have direct measures of OASIS elapse time in these cases, this can deduced by EC-Earth3 elapse time that decreases from 45.7 to 42.3 seconds (512-128-1 -> 512-128-10 configurations) and from 33.2 to 30.3 seconds (800-256-1 -> 800-256-10 configurations). Therefore, it can be concluded that OASIS3 pseudo-parallelisation can be an efficient way to reduce the coupling overhead (which goes from 13.4% to 3% in the 512-128 configuration and from 11% to 1.3% in the 800-256 configuration).

Of course, this way of “hiding” the cost of OASIS3 works only if there is some imbalance of the components elapse time which allows OASIS3 to interpolate the fields when the fastest component waits for the slowest. If the components were perfectly load balanced, then OASIS3 cost, even if lower when OASIS3 is used in the pseudo-parallel mode, would be directly added in the coupled model elapse time.

APPENDIX B – ACRONYMS

CERFACS : Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (France)

CNRS : Centre National de la Recherche Scientifique (France)

CCRT: Centre de Calcul Recherche et Technologie

COSMO-CLM: Climate Limited-area Modelling Community Model in Climate Mode

D&T: Driver & Transformer (OASIS4)

DKRZ: Deutsches KlimaRechenZentrum (Hamburg, Germany)

ECHAM: ECMWF HAMburg (atmospheric general circulation model developed at the Max Planck Institute for Meteorology)

ECMWF : European Centre for Medium range Weather Forecast

ESM(s) : Earth System Model(s)

ETH : Eidgenössische Technische Hochschule (Zurich)

GCM: General Circulation Model

GEMS: Global and regional Earth-system (Atmosphere) Monitoring using Satellite and in-situ data (http://gems.ecmwf.int)

I/O: Input/Output

IPSL : Institut Pierre Simon Laplace (France)

KNMI : Koninklijk Nederlands Meteorologisch Instituut

LMD: Laboratoire de Météorologie Dynamique

MPI: Message Passing Interface

MPI-M: Max Planck Institute for meteorology

MPI-OM: Max Planck Institute for meteorology Ocean general circulation Model

NEMO: Nucleus for European Modelling of the Ocean (**http://www.locean-ipsl.upmc.fr/NEMO/**)

OASIS: Ocean Atmosphere Sea Ice and Soil coupler

OPA: IPSL ocean model

PRISM: Program for Integrated Earth System Modelling (http://prism.enes.org/)

PSMILe: Prism System Model Interface Library

SMHI : Swedish Meteorological and Hydrological Institute

WRF: Weather Research and Forecasting Model

XML: Extensible Markup Language