Parallel ocean-atmosphere coupling: from OASIS3 to OASIS4

Sophie Valcke[1], Eric Maisonnave[1], Laure Coquart[1]

[1] Sciences de l'Univers au CERFACS, CERFACS/CNRS URA 1875, 42 Avenue G. Coriolis, 31057 Toulouse Cedex 01, France

Abstract

OASIS est un logiciel développé depuis 1991 au CERFACS qui permet de coupler des codes numériques représentant les diverses composantes du système climatique terrestre (océan, atmosphère, glace de mer, surfaces continentales, …), c'est-à-dire d'échanger et de transformer de l'information (des « champs de couplage ») de façon synchronisée à l'interface de ces composantes. Aujourd'hui, OASIS est utilisé par plus de 30 groupes de recherche en modélisation climatique à travers le monde.

Deux versions d'OASIS sont actuellement disponibles: OASIS3, résultat de plus de 15 années de développement, et OASIS4, nouvelle version complètement parallèle autant au niveau de l'échange que de la transformation des données. La parallélisation du couplage est un passage obligé pour l'exécution de modèles couplés sur des plates-formes de calcul à architecture massivement parallèle.

Nous détaillons ici les caractéristiques techniques des deux versions du coupleur. Nous présentons également quelques couplages à plus hautes résolutions réalisés avec OASIS3 sans que la parallélisation partielle d'OASIS3 n'entraine de goulot d'étranglement ainsi que des mesures d'efficacité et de scalabilité de l'algorithme multi-grilles d'OASIS4.

History

In 1991, CERFACS started research in climate modelling with the objective to perform the technical assembling of an ocean General Circulation Model (GCM), OPA developed by the Laboratoire d'Océanographie Dynamique et de Climatologie (LODYC), and two different atmospheric GCMs, ARPEGE and LMDz developed respectively by Météo-France and the Laboratoire de Météorologie Dynamique (LMD).

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 and used in a 10-year coupled integration of the tropical Pacific Terray1995]. At the time, the communication, i.e. the exchange of coupling fields, was ensured with CRAY named pipes and ASCII files. Between 1995 and 2000, alternative communication techniques based on the Parallel Virtual Machine (PVM, http://www.csm.ornl.gov/pvm/), UNIX System V Interprocess Communication (SVIPC), NEC SX global memory communication and on the Message Passing Interface (MPI) were introduced while the community of users was steadily and rapidly growing in Europe but also in Australia and in the USA. From 2001 until 2004, the development of OASIS benefited from an important support from the European Commission in the framework of the PRISM project (http://prism.enes.org)} [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, including in particular a new Application Programming Interface (API) 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 the concepts of portability and flexibility that made the success of OASIS3.

Today, both the widely used OASIS3 coupler [Valcke06b] and the new fully parallel OASIS4 coupler [Redler2010] are available. As detailed below, OASIS is used today by about 35 different climate modelling groups in Europe, Australia, Asia and North America.

Both OASIS3 and OASIS4 are portable set of Fortran and C routines which, at run time, act as a separate Driver & Transformer executable performing driving and regridding tasks, and as a model interface library, the PSMILe, that needs to be linked to and used by the component models. In the next paragraphs, technical details about the coupler functionality are provided, emphasizing the similarities and differences between OASIS3 and OASIS4.

Coupling configuration

At run time, the OASIS Driver first 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 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. During the run, the Driver-Transformer executable and the component model PSMILes perform appropriate exchanges based on this configuration.

With OASIS3, the configuration information is contained in a text file following a specific format while with OASIS4 it is provided in Extensible Markup Language (XML, http://www.w3.org/XML/) files. A Graphical User Interface (GUI) facilitates the creation of those XML files.

Process management

In a coupled run using OASIS3 or OASIS4, the component models generally remain separate executables with main characteristics, such the general code structure or the memory management, untouched with respect to the uncoupled mode. The user has to take care that the component models coherently define some global parameters such as the total run duration, the calendar, etc.

OASIS supports two ways of starting the executables of the coupled application. If a complete implementation of the MPI2 [Gropp1998] is available, only the OASIS Driver has to be started by the user. All remaining component executables are then launched by the OASIS Driver at the beginning of the run using the MPI2 MPI_Comm_Spawn functionality. If only MPI1 [Snir1998] is available, the OASIS Driver 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, OASIS needs to recreate a component model communicator that must be used by the component model for its own internal parallelisation. 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.

Communication: the OASIS PSMILe library

To communicate with other component models or to perform I/O, a component model needs to call few specific OASIS PSMILe routines. The PSMILe API function calls for both OASIS3 and OASIS4 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 OASIS4 PSMILe Application Programming Interface (API) was kept as close as possible to OASIS3 PSMILe API; this should ensure a smooth and progressive transition between OASIS3 and OASIS4. The main difference between OASIS3 and OASIS4 PSMILe API remains in the grid definition; while with OASIS4 the description of the local partition of the grid covered by the local process is mandatory, with OASIS3 the global grid definition can either be provided by the component master process through the PSMILe API or by the user in a NetCDF file constructed before the run.

For both OASIS3 and OASIS4, 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 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 argument and the sending/receiving is actually performed only if the date corresponds to a time at which it should be activated, given the field coupling or I/O frequency indicated by the user in the configuration file; a change in the coupling or I/O frequency is therefore also totally transparent for the component model itself.

Both OASIS3 and OASIS4 PSMILe support 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 Transformer which gathers the whole coupling field, transforms or regrids it, and redistributes it to the target component model processes. With the OASIS4 PSMILe, the communication is more efficient as the communication pattern between the source and target processes is based on the intersection of the local domains covered by each source and target component process; therefore, only the useful part of the coupling field is extracted and transferred, either directly between the models (when only repartitioning is needed) or via the parallel Transformer (when repartitioning and regridding are needed).

Coupling field transformation and regridding in OASIS3

For each coupling exchange, the OASIS Transformer receives the source coupling field (or part of it) from the source model, performs the transformations and regridding needed to express the source field on the grid of the target model, and sends the transformed field (or part of it) to the target model.

As stated above, within OASIS3, the transformation is done by the Transformer mono-process executable on the whole coupling field after its gathering in the Transformer memory. The neighbourhood search, i.e. the determination for each target point of the source points that contribute to the calculation of its regridded value, and the corresponding weight calculation is done by the Transformer at the beginning of the run considering the whole source grid. Different transformations on 2D coupling field in the Earth spherical coordinate system are available for grids that are regular in longitude and latitude, stretched, rotated, Gaussian reduced, and unstructured:

- time accumulation or averaging
- correction with external data read from a file
- linear combination with other coupling fields,
- addition or multiplication by a scalar
- nearest-neighbour, Gaussian-weighted, bilinear, bicubic 2D interpolations
- 2D conservative remapping (i.e. the contribution a each source cell is proportional to the fraction of the target cell it intersects;
- user-defined regridding (the weights and addresses are pre-defined by the user in an external file)
- global conservation
- creation of subgrid scale variability (when regridding from low to high resolution)

The interpolations and conservative remapping are taken from the Spherical Coordinate Remapping and Interpolation Package (SCRIP) library. OASIS3 also supports interpolation of vector fields with the projection of the two vector components in a Cartesian coordinate system, interpolation 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.

With the last release of OASIS3, it is now possible to use more than one OASIS3 Transformer executable in a coupled system, each executable treating a subset of the complete coupling fields; the result is a pseudo-parallelisation of OASIS3 Transformer on a field-per-field basis.

Coupling field transformation and regridding in OASIS4

During the run, the OASIS4 parallel Transformer manages the transformation and regridding of 2D or 3D coupling fields. The Transformer performs only the weight calculation and the regridding *per se* while the neighbourhood search is performed in parallel in the source PSMILe; this ensures that only the useful part of the coupling field is extracted and transferred.

In a simple implementation of a neighbourhood search algorithm, such as the one used in OASIS3, the M ``neighbours'' of a target grid point can be identified by comparing its distance with all N source points. While such an approach of the order (M x N) is still justified for relatively small problem sizes, it may become very costly for high resolution grids. For OASIS4, a more efficient algorithm is implemented.

In a first step, envelopes of the grid partitions residing on each process are defined, exchanged between source and target processes and intersections are identified. For each intersection, the list of target points is sent to the source process; a grid hierarchy, similar to what is used by a multigrid algorithm, is established on the source side with a refinement factor of two. In OASIS4, this hierarchy is used to identify the source cell containing the projection of each target point. From this step onwards, the identification of the neighbours involves only local operations in the grid point space. The great advantage of the “multigrid" algorithm is that it is of the order of (M x logN) and thus only weakly dependent on the source grid size; e.g. only one additional level would be introduced in the source grid hierarchy if the source grid size was doubled.

When the source grid is partitioned, some neighbours of a target grid point which projection is contained into one source partition near its border can in fact be located on an adjacent partition. OASIS4 performs this additional search step, called the parallel `"global" search, into which the source neighbours are also searched in adjacent partitions. If neighbour source points are found in adjacent partitions, the full information about those source points is returned to the process that has initiated the search. The global search therefore ensures that the regridding result is independent of the source partitioning.

At the end of the PSMILe neighbourhood search, each source process holds different lists, each list containing the information about the target points located in the intersection of a target process domain with its local domain and about the source neighbour points needed for the regridding of these target points.

These lists are initially equally distributed over the Transformer processes, resulting in an effective parallelisation of the Transformer over the lists. During the simulation time stepping, the OASIS4 parallel Transformer can be assimilated to an automate that reacts to what is demanded by the different component model PSMILes. During the exchange phase, each Transformer process receives from the source PSMILe the grid point field values (transferred from the source component with a prism_put call, calculates the regridding weights if it is the first exchange, and applies the weights. The data are sent upon request from the respective target process (i.e. when a prism_get is called in the target component code). The OASIS4 Transformer therefore acts as a parallel buffer into which the transformations take place.

In OASIS4, the following transformations 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 yet):

- 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)
- global conservation

The parallel global search is implemented for all grids supported and for all interpolations. As in OASIS3, 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.

OASIS4 does not support vector interpolation. For an exact interpolation, 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.

User community

Since the first version released in 1993 and used at CERFACS, Météo-France and IPSL in France, the number of OASIS users steadily increased and reaches today a community of about 35 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.

The current user community of OASIS4 is of course much smaller but use of OASIS4 has already shown promising results in different configurations. 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 [Flemming09]. 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. Global ocean-atmosphere coupled models are also being currently set-up with OASIS4 at the UK MetOffice, the MPI-M, the ETHZ (Swiss Federal Institute of Technology) and at CERFACS.

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 pseudo-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:

- 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.
- 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 evry 2 hours. The coupled system uses up to 2191 cpus, with 2048 for LMDz, 120 for NEMO, and 23 for OASIS3
- 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 in parallel and not sequentially. We can observe here that OASIS 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 which 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.

OASIS4 performances

Even if OASIS4 is currently being used in real coupled models (see paragraph “User Community” above), we have no real measure of performance yet. We therefore use here a simplified case to perform a first test of the PSMILe library and Transformer 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 consider 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.

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 been 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 use 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.

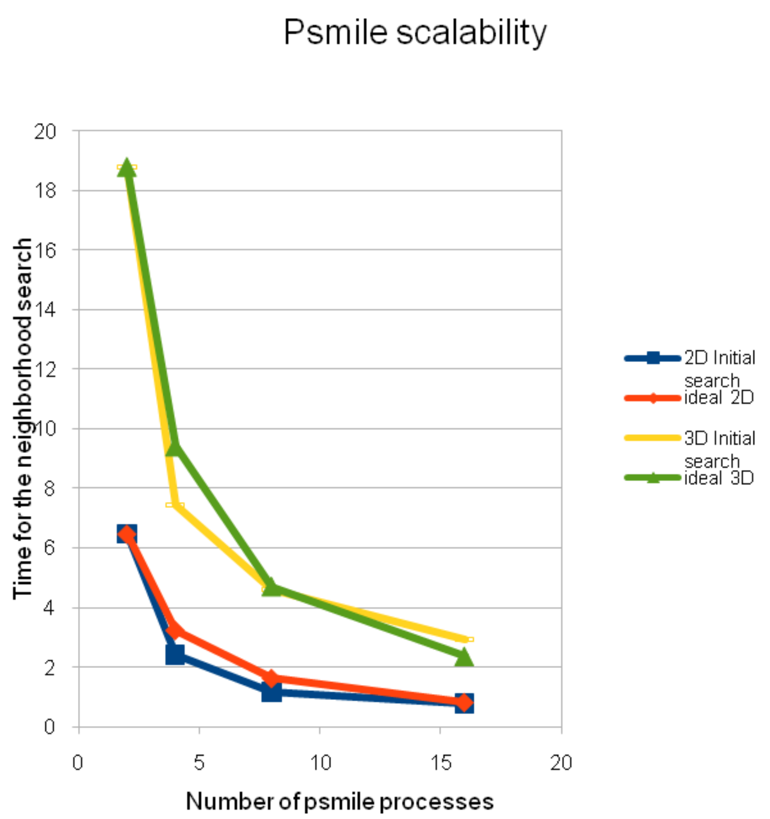
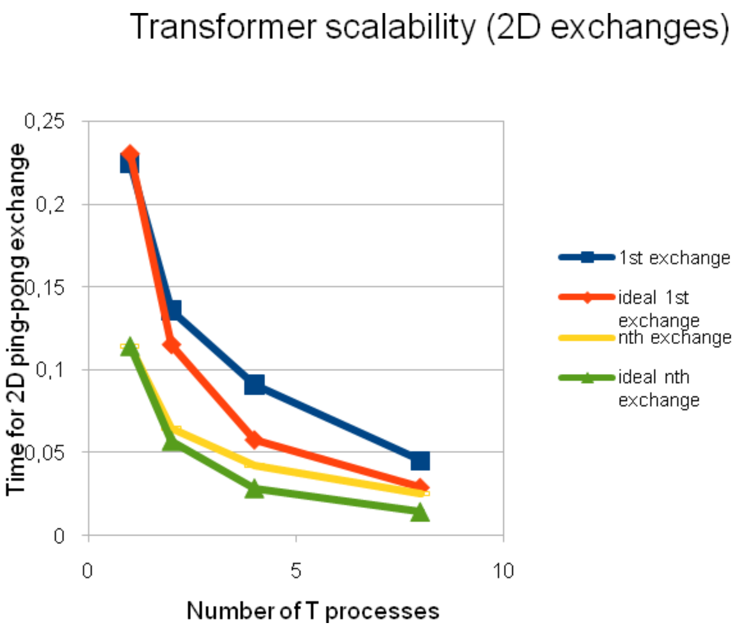


Figure 1 – Time to perform the neighbourhood search as a function of the number of PSMILe processes for the 2D and 3D cases. Ideal curves suppose that the time decreases by a factor X when the number of processes is increased by the factor X.

Figure 1 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 1, 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.

Scalability of the Transformer

Figure 2 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 (top) and the 3D case (bottom).



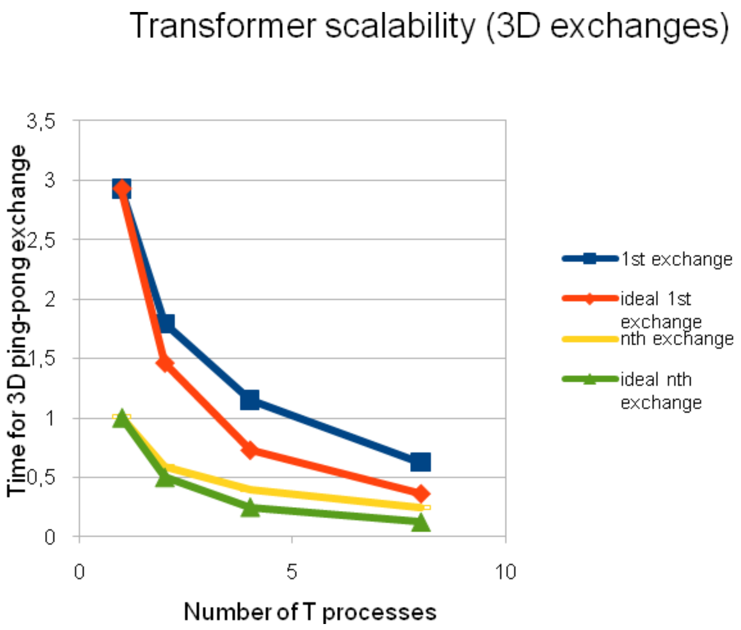


Figure 2 – Time to perform the neighbourhood search as a function of the number of PSMILe processes for the 2D and 3D cases. Ideal curves suppose that the time decreases by a factor X when the number of processes is increased by the factor X.

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 Transformer shows a good scalability.

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

Performance of the multi-grid search and parallel regridding

To evaluate the performance of OASIS4 and in particular the efficiency of the 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.

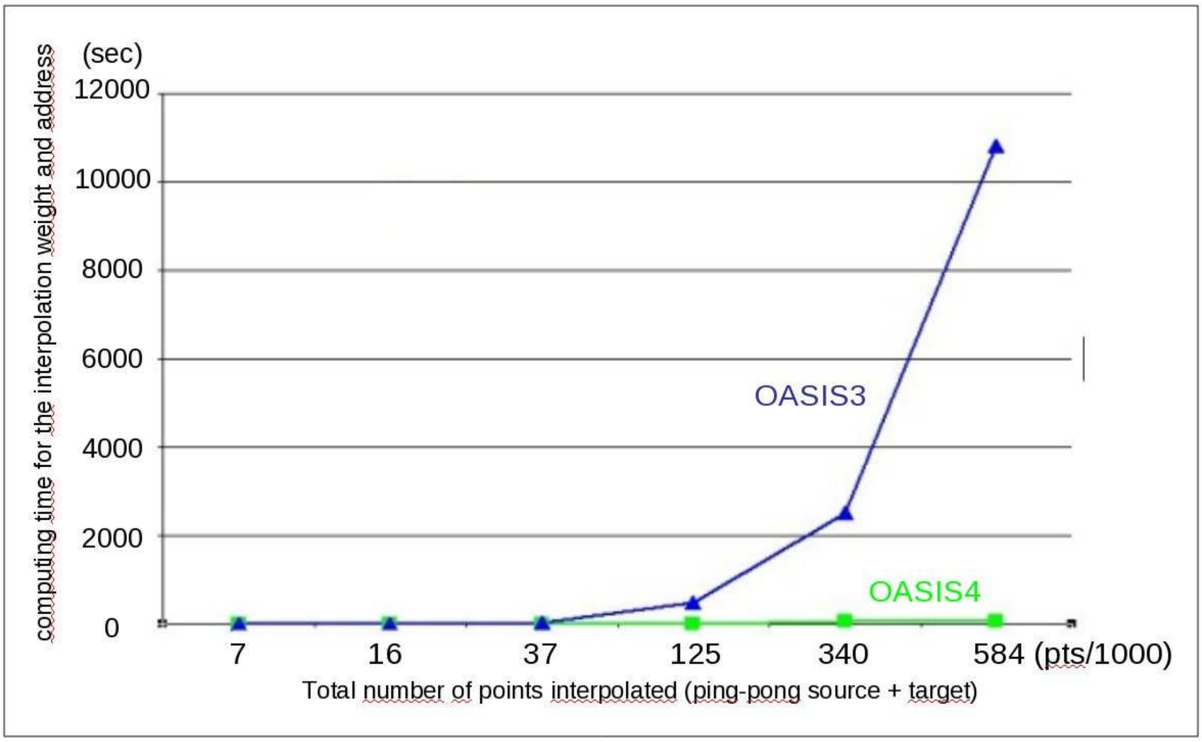


Figure 3 – Time for the search and the 1st exchange as a function of the total number of points interpolated for the atmosphere and the ocean during the ping-pong exchange for OASIS3 (in blue) and for OASIS4 (in green)

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 Transformer and the components were running with one process each. The numbers provided in Figure 3 give the time for the search and the 1st exchange (again this measure was almost the same for both components). We provide 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 was observed that OASIS3 is about two times slower than OASIS4 (but this is not visible on Figure 3). 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(N2) for OASIS3 where as it increases only with O(N) for OASIS4. This clearly demonstrates the benefit of the multi-grid neighbourhood search when compared to a classical search and the increased general performance of OASIS4 over OASIS3 even in this simple non-parallel case.

Conclusions and perspectives

The OASIS coupler is software allowing synchronized exchanges of coupling information between numerical codes representing different components of the climate system. At run-time, it acts as a separate application that performs the interpolation of the coupling fields and as a communication library linked to the component models. Two versions of the coupler are currently available. OASIS3 is the direct evolution of the 2D coupler developed since about 20 years at CERFACS. OASIS4 is a newer fully parallel 3D coupler developed in collaboration with NEC Laboratories Europe - IT Research Division (NLE-IT) and the French Centre National de la Recherche Scientifique (CNRS). OASIS3 is stable and well debugged but it is more limited than OASIS4, which however still needs some validation, especially in the fully parallel cases. The OASIS4 PSMILe Application Programming Interface (API) was kept as close as possible to OASIS3 PSMILe API; this should ensure a smooth and progressive transition between OASIS3 and OASIS4 use in the climate modelling community.

Within the framework of funded projects work continues to establish comprehensive services around OASIS through a portal offering documentation, user guides, tutorial, FAQs, user forum and tips for best practices. Last but not least, the software development is continued to add new functionality and extend existing one. On example of such an initiative is the InfraStructure for the European Network for Earth Systm Modelling (IS-ENES), a 4-year project started in March 2009 that brings about 90 person-months of funding for OASIS development and user support, and into which a fruitful collaboration with the Deutsches Klimarechenzentrum GmbH (DKRZ) is currently taking place. In addition to the tasks already mentioned above, within the IS-ENES project personal user support is provided to the climate modelling community to assemble new models coupled with the OASIS3 or OASIS4 coupler, or migrating from OASIS3 to the fully parallel OASIS4.

Currently, CERFACS and CNRS are committed to support the development and maintenance of the OASIS software. However, CERFACS and CNRS permanent resources devoted to the OASIS development are most probably undersized given OASIS large user community and the always evolving complexity of the computing platforms used in climate modelling. Therefore CERFACS current objective is to establish an official Memorandum of Understanding between the largest institutions using OASIS into which each partner would engage in spending some permanent resources on OASIS.

OASIS capitalizes about 35 person-year of work and is used by about 35 modelling groups. The average of 1.0 person-year/group is certainly much less than the time it would have taken for each group to develop its own coupler. Therefore, OASIS is and hopefully will remain for the coming years thanks to the planned MoU a great example of successful community software.

Acknowledgements

Part of this work has been funded by the EU FP7 IS-ENES project (Contract GA No: 228203). We would like to thank especially Uwe Fladrich for his nice welcome during Eric Maisonnave's stay in SMHI when tests done on EC-Earth3 reported above were realised, and René Redler for its active collaboration on OASIS4 past and current developments and scalability tests.

References

- [Terray1995] Terray, L., O. Thual, S. Belamari, M. Déqué, P. Dandin, C. Lévy, and P. Delecluse. Climatology and interannual variability simulated by the arpege-opa model. *Climate Dynamics*, 11:487--505, 1995
- [Redler2010] R. Redler and S. Valcke, 2010. OASIS4, A Coupling Software for Next Generation Earth System Modelling. *Geosci. Model Dev.,* 3, 87-104.
- [Flemming2009] J. Flemming, A. Inness, H. Flentje, V. Huijnen, P. Moinat, M. G. Schultz, and O. Stein, 2009. Coupling global chemistry transport models to ECMWF's integrated forecast system, GMDD, 2, 763-795
- [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.
- [TR-Valcke2006b] S. Valcke, 2006. OASIS3 User Guide (prism_2-5), Technical Report TR/CMGC/06/73, PRISM Report No 2, CERFACS, Toulouse, France. 60 pp
- [Balaji2001] Balaji, 2001: Parallel Numerical Kernels for Climate Models, ECMWF TeraComputing Workshop 2001, World Scientific Press, Reading.
- [Jones1999] Jones, P.: Conservative remapping: First- and second-order conservative remapping, Monthly Weather Review, 127, 2204–2210, 1999.
- [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.
- [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.