IV International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2011
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

O-PALM: AN OPEN SOURCE DYNAMIC PARALLEL COUPLER

ANDREA PIACENTINI [*], THIERRY MOREL [*], ANTHONY THÉVENIN [*] AND FLORENT DUCHAINE [*]

[*] Centre Européen de Recherche et Formation Avancée en Calcul Scientifique (CERFACS)
42, av. G. Coriolis
31057 Toulouse Cx 1, France
e-mail: palm@cerfacs.fr, www.cerfacs.fr/globc/PALM_WEB

Key words: Dynamic coupling, Parallel computing, Graphic User Interface, Open-Source, Multi-physics, Data Assimilation.

Abstract. Since 1996 CERFACS has been developing the PALM parallel coupler, which is currently used for more than 50 research and industrial projects ranging from operational data assimilation to multi-physics modelling, from climate change impact assessment to fluid and structure interactions. It can be defined a dynamic coupler for its ability to deal with situations where the component execution scheduling and the data exchange patterns cannot be entirely defined before execution. Under the name O-PALM, it is now open source (LGPL license). This document provides some highlights on the design of PALM and on the main implementation choices and a brief description of some representative applications.

1 INTRODUCTION

CERFACS has a long and well established experience in parallel computing and in model coupling. In particular, the OASIS coupler [1] is a reference tool for coupled climate simulations [2,3] since the 90's. For this reason, CERFACS was chosen by the MERCATOR operational oceanography French project [4] for the set-up of the software framework to be used for the implementation of a new operational data assimilation suite as a coupling between existing and under development models and tools.

Section 2 will detail the rationale for the design and implementation of a dynamic parallel coupler to answer this commitment and will introduce other applications that in turns benefited from the use of a dynamic coupler and contributed to further define the O-PALM design.

Section 3 will sketch the main technical choices that were adopted to implement a very flexible and user friendly coupler, still granting high performances and portability.

Section 4 will briefly list the new challenges that a coupler has to face when preparing the next generations of coupled models on massively parallel and exascale machines.

Finally, Section 5 will present some representative applications currently using O-PALM in research and production.

2 DESIGN

O-PALM is a dynamic coupler in the sense that it can be used to implement a coupling where the component execution scheduling and the data exchange patterns cannot be entirely defined before execution.

The historical reason for this feature dates back to 1996 when the MERCATOR operational oceanography project faced the problem to set-up a new operational suite with Data Assimilation for research and operations in an evolving configuration.

Data assimilation can be roughly defined as a collection of techniques aiming to improve numerical models skills by the use of observational data. They are based on computationally expensive algebraic algorithms involving a model, observation treatments, and the statistical characterisation of the errors on both sides. At that time, in MERCATOR, the choice of the model configuration was not yet finalized, there were several candidate assimilation methods to test, there were different kinds of observations to handle and the same system should have been used for research and operations. All these needs of flexibility lead to the implementation of the assimilation suite as a coupling between model, observations handling, error statistics and algebra instead of hard-coding data assimilation routines in the model, or vice-versa [5].

Some data assimilation algorithms are based on an iterative minimization: this implies the repeated execution of the tasks and the total number of iterations is not necessarily known beforehand. Moreover, in some configurations some tasks are activated only if some observations are available at run-time.

This specific requirement imposed to conceive a coupler of independent parallel codes capable to deal with complex coupling algorithms allowing for the conditional and/or repeated execution of the coupled components. The main goals and constraints were user friendliness, modularity, portability and high performances on parallel computers.

Existing couplers were not a suitable choice for the lack of the dynamic aspects. This lead, after a thorough feasibility study [6], to the design of a new MPMD dynamic parallel coupler, based on the MPI message passing and process management standard library [7,8].

In our definition, a dynamic coupler has to fulfil three main requirements:

- process management: this means that the coupler has to be able to start and synchronise the tasks and to handle algorithms with loops and conditional switches
- buffered communications: in order to grant full flexibility, avoiding deadlocks dependencies on the production and reception order, at least the production side of a communication has to be non blocking. This requires the explicit handling of a storage space for pending communications. This feature allows for some extra possibilities, such as the linear combination of cumulated fields and the explicit permanent storage of objects that are to be repeatedly received
- object versioning: the flexible use of a temporary storage space for parallel communications requires special care to grant the coherency of the stored global objects. The Last In Only Out paradigm is adopted: every new version of an object replaces the previous ones. Nevertheless, for parallel communications, we count a new version of an object only when all the processes of the producing code have provided

their contributions. For loosely synchronised codes, it implies the introduction of stamps to keep track of what version of an object new contributions belong to.

The same way, in a parallel coupling, a coupler has to deal with two levels of parallelism:

- concurrent tasks parallelism: independent tasks can run concurrently on separate sets of processes. The coupler has to deal with the concurrent execution, to establish all the needed intercommunication contexts and to grant synchronisation.
- distributed coupled codes: as a second level of parallelism we account for the inner parallelism of the coupled codes, mostly related to data distribution. The coupler has to grant private and robust intracommunication contexts and, most important, to be able to manage the data exchanges between sets of processes, including the remapping between codes with different distributions of the same physical objects.

Since one of the main aims of coupling is the reuse of legacy codes, we tried to reduce the intrusiveness of the coupling instructions in the source codes. For this reason we adopted the so-called end point communication paradigm: the producer of an object does not know anything about the recipients (if any) and the other way round. The coupler makes the matching.

In order, once more, to minimize the interventions in the codes, we defined a reduced set of multi-language API calls, complemented by a very detailed Graphic User Interface: most information - such as the coupling algorithm, the communication patterns and the parallel distributions - is easily described in the graphic interface. The changes in the code have minimal impact and, because of the use of the one-sided communications paradigm, they are independent of the specific coupling algorithm.

O-PALM can thus be interfaced to F77, F90, C or C++ compiled codes or to the main interpreted languages such as Python, Perl, Java, Tcl/Tk, Octave and also to black-box pre-compiled executables that can be linked at run-time against dynamic libraries.

The last constraints that drove the design of O-PALM are related to its uses and diffusion. The operational usage imposes robustness and high performances. This not only determined some implementation choices, but it also lead to the integration in O-PALM of a real-time monitor, allowing to display in the graphic user interface the status of the execution while running and of a performance analyser that works on trace files and helps tuning and optimising the coupled application. For research applications there are other criteria, such as portability, that imposed to rely on standard coding and message passing techniques) and user friendliness. The latter not only drove the design of the Graphic User Interface, but it lead also the the introduction of an algebra toolbox providing a palette of predefined generic algebraic operations ranging from BLAS to parallel linear algebra solvers and minimisers that can be coupled to any other user defined code.

Since then, the range of O-PALM applications has largely extended beyond data assimilation and it is the coupler of choice in a number of multi-physics coupling when dealing with flexible configurations [9]. Most relevant applications are in computational fluid dynamics for fluid-structure interactions modelling or for automatic shape optimisation computations or in event driven couplings in soil surface modelling [10]. New applications carried new requirements for the design of O-PALM future versions, but also some contributed developments. For this reason, the coupler has become open-source at the beginning of 2011. This is the most suitable environment to deal with the new challenges that massively parallel machines and exascale computing will carry into high performances scientific and engineering computing.

3 IMPLEMENTATION

The coupler implementation went through several steps.

At the very beginning of the project, when the MPI2 standard [11] was recently published, but hardly any complete and robust implementation was available, we implemented an MPI1 emulation based on a pool of idle processes, released under the name of PALM_RESEARCH, later changed into PALM_SP. It was dedicated to functional tests, but in practice it proved to be very effective in some cases and it still used for some full size applications. Some interesting features of this first implementation could now be seen as possible optimisations under some conditions and will be hopefully reintroduced in the current O-PALM version in a near future.

In 2003 we released the first fully MPMD version of O-PALM under the name PALM_MP. It was based on the MPI2 process management and communication layer. The main components of PALM_MP are

- the scheduler that handles the process management and the execution of the coupled components accordingly to the algorithm described in the user interface. PALM can schedule several parallel codes to run concurrently to perform independent tasks if enough resources are available. Since starting an independent executable always causes a overhead, PALM offers the option to merge into a single executable the coupled components that are started in a sequence.
- the optimised communication scheme managed by a driver that takes care of the data transfer between parallel programs. This is one of the most evolved components of PALM and handles very complex communication patterns with some very practical features, such as the remapping of objects exchanged by parallel codes with different distributions, the selection of object subsets entirely from the user interface, the presence of an explicitly managed permanent repository for objects to be repeatedly received.

Since then the coupler has been constantly enhanced and optimised.

With respect to PALM_MP, the current O-PALM release offers

- the possibility to interface commercial black-box codes (such as Fluent, Abaqus MSC/MARC) by the use of external dynamic libraries and/or a socket based layer
- the possibility to interface interpreted languages such as
- the implementation of a simplified working mode entirely compliant with the MPI-1 library
- the optimisation of repeated well synchronised communications that don't require the intervention of the driver
- the enhancement of the parallel algebra toolbox that is soon going to include the

CWIPI interpolation library from ONERA for the grid to grid remapping.

More technical details on the implementation go beyond the scope of this paper. A detailed source of information is the web site dedicated to the O-PALM coupler http://www.cerfacs.fr/globc/PALM_WEB.

The Graphic User Interface, called PrePALM, is a portable Tcl/Tk application (Fig. 1). It deserves some words because it is the part of the coupler that users interact with most of the time. The relevant features of the coupled components are described in identity cards that do not depend on the specific coupling algorithms. The user describe the execution scheduling, the parallel sections, the data exchange patterns and the algebraic treatments, entirely from within the user interface. It ends up providing the input file for the coupler executable itself and the source code for the wrappers of the coupled component that take entirely care of the set-up of the communication context with no need of change in the components sources.

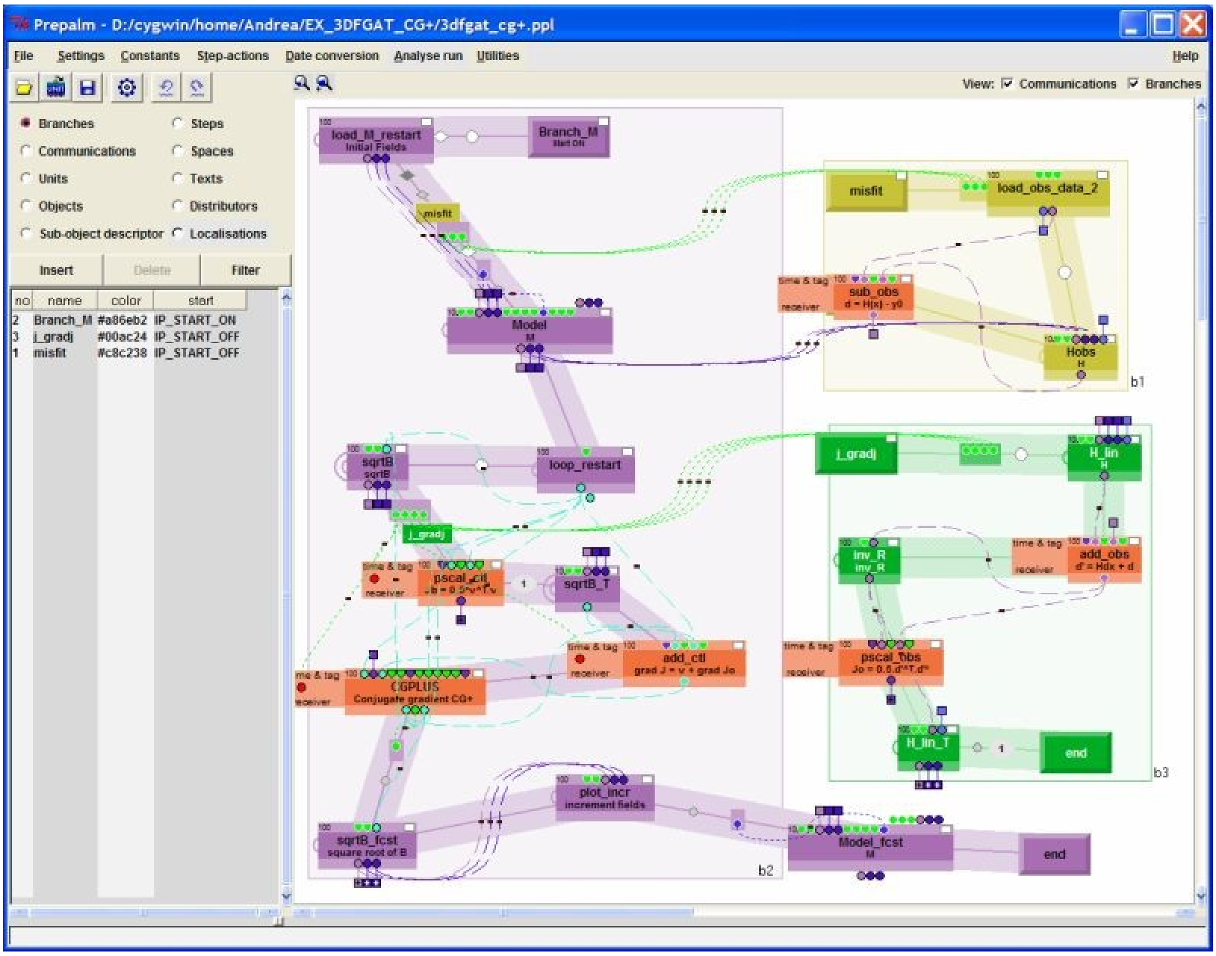

Figure 1: The PrePALM Graphic User Interface

The same graphic tool can be used at run-time to monitor the simulation status, with colour codes to distinguish between components running, waiting for resources or not yet scheduled and with counters for the completed executions and the performed data exchanges. The same tool again provides post-mortem some statistics on the memory and CPU time resources used by the components (Fig. 2.).

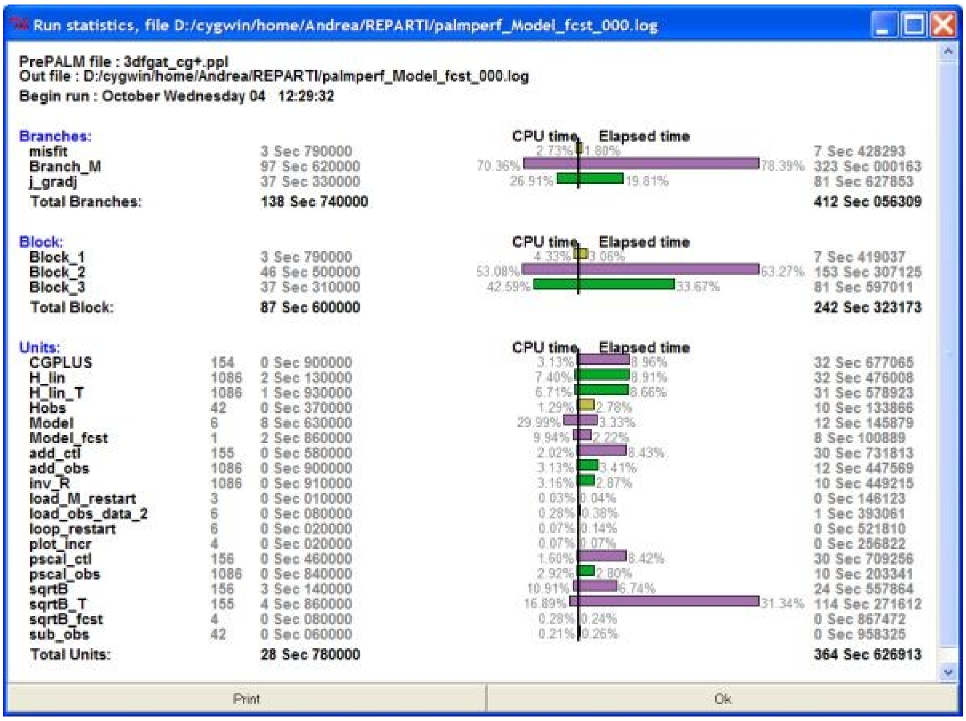

Figure 2: The PrePALM Graphic User Interface

4 FUTURE DIRECTIONS

Among the most important technical challenges for the evolution of O-PALM towards exascale applications, there is the search for the best trade-off between a centralised and a fully distributed approach. If on one side process management and monitoring is a key issue for dynamic coupling, it imposes an overhead and a risk of bottlenecks for massively parallel applications. Optimisations bypassing the O-PALM scheduler and launcher are under study, but they have to keep the capability of reorganizing the layout of the application in case, for instance, of automatic load balancing or adaptive meshes.

The same considerations apply for the communication handling: to obtain very effective parallel communications on massively parallel configurations, we'll have to look the best compromise between flexibility and monitoring on one side and performances on the other.

Furthermore the new directions in high performances computing introduce some further constraints, such as the capability to reorganise the coupling layout around self-tuning applications, changing their configuration at run-time or the capability to implement a resilient parallel coupling on a very high number of processors.

Finally, some application domains require specific treatments. It is the case of the physical constraints (e.g. mass or energy conservation) of the grid-to-grid interpolations for climate modelling applications. In such a case, the use of O-PALM and the CWIPI interpolation library (with specific enhancements) has to be thoroughly studied and evaluated on test cases of increasing complexity and size.

The open source distribution of O-PALM is the most suitable environment to accept collaborations and contributions on the coupler development on all these topics.

5 SOME APPLICATIONS

Current O-PALM applications largely go beyond data assimilation and cover many fields of multi-physics coupling ranging from oceanography to hydraulics, from hydrology to agronomy, from aeronautics to space engineering and so on.

Some of them are particularly representative of the advantages coming from the dynamic coupling and from the user friendliness of the API's and of the user interface.

For instance we could mention the use of O-PALM for the coupling of an adaptive 2D surface biosphere model to different parallel atmosphere circulation limited area models. In such a case, not only the dynamic features of O-PALM can easily take into account the adaptive model, but also the compact syntax used to describe data exchanges allows for a quite generic implementation with different atmosphere models [12].

Some full size, near real time applications, like the operational ocean data assimilation and forecast suite of the MERCATOR operational oceanography centre or the air quality data assimilation and forecasting system Valentina, based on the MOCAGE chemistry and transport model [13], provide a very satisfactory test bench for the O-PALM performances in large scale parallel applications.

We should also mention the recent use of O-PALM for the implementation of a demonstrative data assimilation suite based on a 1D hydraulic model used in flood forecasting. The graphic algorithm representation proves to be a very useful pedagogical tool. Furthermore, the generic formalism allows for the application of the demo suite to real life applications with no changes in the code lines [14].

To give a detailed, application oriented example of the use of O-PALM in real size problems, we present here a study in a relevant computational fluid dynamics research domain: the determination of heat loads that is a key issue in gas turbines conception, because wall temperatures and heat fluxes are a major constraint in the design of combustors and turbine blades. Indeed, the life duration of turbine components directly depends on the wall temperature and therefore designers imperatively need an accurate prediction tool. Numerical simulations of the thermal interaction between fluid flows and solids is therefore of primary interest. The complex flows observed in the turbine environment are much better predicted in the Large Eddy Simulations (LES) framework, especially when considering thermal effects.

To simulate a cooled turbine blade it is necessary to couple a LES solver and a heat transfer code within solids [15,16]. The LES solver used is the AVBP code [17,18,19,20], which solves the full compressible Navier-Stokes equations on unstructured meshes, using a cell-vertex/finite element approximation and a Taylor-Galerkin weighted residual central distribution scheme. The calculation of thermal diffusion in solids is performed with the code AVTP, solving the classical heat equation [16]. AVTP also uses unstructured meshes and is advanced with a first-order explicit forward Euler scheme.

The study simulates a cooled blade of the T120 cascade, which was designed by Rolls Royce Deutschland for the European project AITEB [21]. Experiments were conducted in the High-speed Cascade Wind Tunnel of the Institute of Jet Propulsion of Aachen [22]. The highly-loaded high-pressure turbine airfoil of the T120 cascade was designed to have a large separation on the pressure side. The blade is operated at a Reynolds number of $3.8 \bullet 10^5$ and a Mach number of 0.87, based on the exit velocity and the chord. The film cooling device of the T120D blade is composed of three holes located on the pressure side, repeated in the spanwise direction to form a pattern of jet rows (Fig. 3-a). The computational domain covers one cooling hole pattern in the spanwise direction, with periodicity boundary conditions. The unstructured mesh is composed of 6.5 millions of tetrahedral elements for the fluid zone, and 600 000 elements within the solid. The skin meshes are the same for the fluid and the solid so that no interpolation error is introduced at this level when CHT is simulated. The converged thermal state is obtained in 10 characteristic solid time scale and requires about 4800 CPU hours. At the converged state, the net heat flux through the blade reaches zero. Figure 3-b shows an instantaneous snapshot of temperature distribution in the fluid and solid domains.

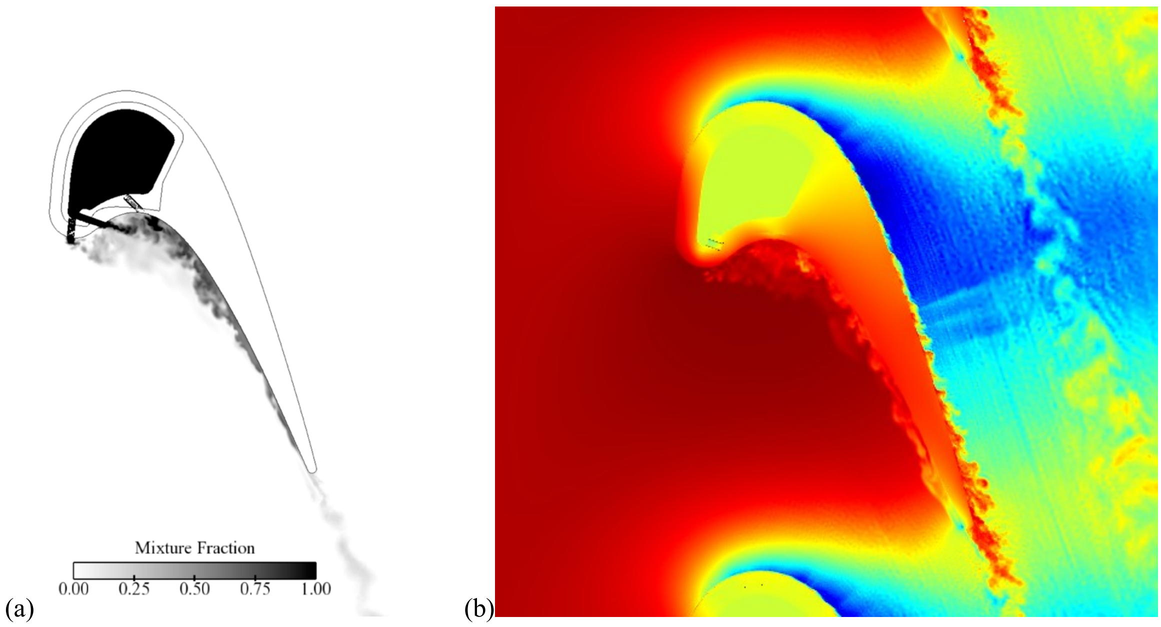

Figure 3: Instantaneous snapshot of cooling air distribution (a) and temperature distribution in the fluid and solid domains (b).

Figure 4 compares experimental and numerical cooling efficiency fields on a 2D plot over the pressure side.

The computation matches the experimental visualization fairly well and evidences the thermal effects of the cooling jets on the vane.

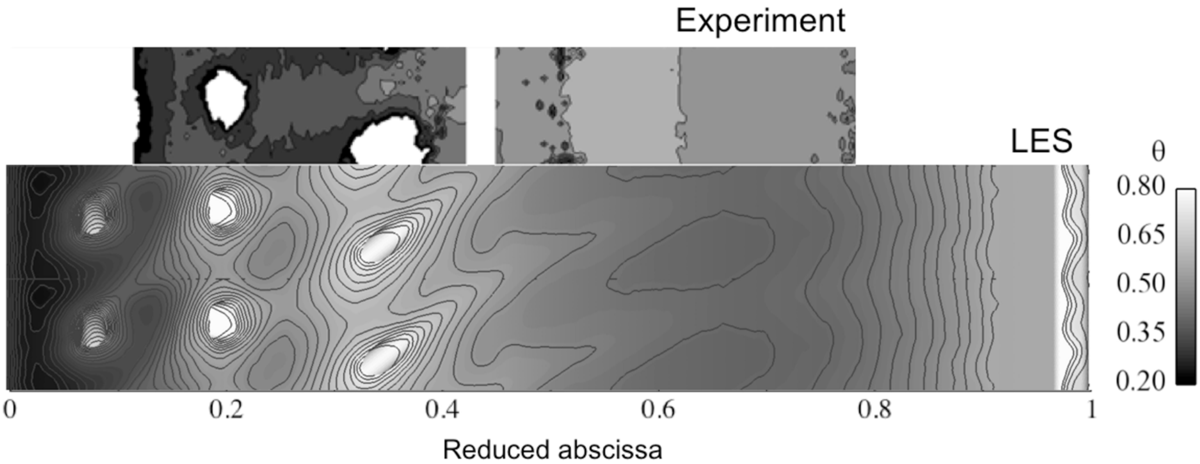

Figure 4: 2D plot of time-averaged cooling efficiency on the pressure side: comparison of experimental results and coupled simulation. The scale of θ corresponds only to the LES field.

6 CONCLUSIONS

The O-PALM coupler has reached a high degree of maturity and stability. It is currently used for more than 50 research and industrial projects ranging from operational data assimilation to multi-physics modelling, from climate change impact assessment to fluid and structure interactions. It implements a dynamic coupling paradigm based on portable and effective technical solutions. It is well suited for the evolution of the current coupling technology towards the exaflop machines of next generations and several research centres and engineering companies are attentively considering this issue. Having become open-source since the beginning of 2011, O-PALM is aiming to be a tool of reference for dynamic and high performances coupling applications and ongoing researches in coupling technology.

REFERENCES

[1] S. Valcke, S. and Redler, R. *OASIS4 User Guide* PRISM–Support Initiative Technical Report No 4 (2006).

[2] Valcke, S., Guilyardi. E. and Larsson, C. PRISM and ENES: A European approach to Earth system modelling**.** *Concurrency Computat.: Pract. Exper.* (2006) **18**(2):231-245.

[3] Valcke, S., Declat, D., Redler, R., Ritzdorf, H., Schoenemeyer, T. and Vogelsang, R. The PRISM Coupling and I/O System. VECPAR'04, Proceedings of the 6th International Meeting, vol. 1 *High performance computing for computational science* (2004).

[4] Bahurel, P. Mercator Océan. Global to regional ocean monitoring and forecasting. In *Ocean weather forecasting: An integrated view of operational oceanography*. Springer (2006):381-395.

[5] André, J.C., Blanchet, I. and Piacentini, A. Besoins algorithmiques et informatiques en océanographie opérationnelle. Le programme Mercator. *Calculateurs Parallèles, Réseaux et Systèmes répartis* (2000) **11**(3):275-294.

[6] The PALM group Etude de faisabilité du projet PALM, *CERFACS Technical Report TR/CMGC/98-50* (1998).

[7] Fouilloux, A. and Piacentini, A. The PALM Project: MPMD Paradigm for an Oceanic

Data Assimilation Software, *Lecture Notes In Computer Science* (1999) **1685**:1423-1430.

[8] Piacentini, A. and The PALM Group, PALM: A Dynamic Parallel Coupler, *Lecture Notes In Computer Science* (2003) **2565**:479-492.

[9] Buis, S., Piacentini, A. and Déclat, D. PALM: A Computational framework for assembling high performance computing applications, *Concurrency Computat.: Pract. Exper.* (2006) **18**(2):247-262.

[10] Duretz, S., Drouet, J.L., Hutchings, N.J., Durand, P., Theobald, M.R., Salmon-Monviola, J., Dragosits, U., Maury, O., Sutton, M.A. and Cellier, P. NitroScape: an integrated model of nitrogen fluxes and transformation at the landscape scale. *Environmental Pollution.* Submitted.

[11] *MPI: A Message-Passing Interface Standard, Version 2.2*, High Performance Computing Center Stuttgart (2009)

[12] Haas, E. and Piacentini, A. Integrated Earth System Modeling - A Coupled Multiscale Bio-, Hydro- and Atmosphere Model for Numerical Process Simulations and Regional Climate Impact Analysis, *Mathematics and Computers in Simulation* (2010) **80**(8) Special Issue ESCO 2008 Conference.

[13] Massart, S., Clerbaux. C., Piacentini, A., Cariolle, D., Turquety, S. and Hadj-Lazaro, J. First steps toward the assimilation of IASI ozone data into the MOCAGE-PALM system, *Atmos. Chem. Phys*. (2009) **9**:5073-5091.

[14] Ricci, S., Piacentini, A., Thual, O., Le Pape, E. and Jonville G. Correction of upstream flow and hydraulic state with data assimilation for flood forecasting, *Hydrol. Earth Syst. Sci. Discuss.* (2010) **7**:1–55.

[15] Duchaine, F., Mendez, S., Nicoud, F., Corpron, A., Moureau, V. and Poinsot, T. Conjugate heat transfer with large eddy simulation application to gas turbine components. *C. R. Acad. Sci. Mécanique* (2009) **337**(6-7):550-561.

[16] Duchaine, F., Corpron, A., Pons, L., Moureau, V., Nicoud, F. and Poinsot, T. Development and assessment of a coupled strategy for conjugate heat transfer with large eddy simulation: Application to a cooled turbine blade. *Int. J. Heat Fluid Flow* (2009) **30**:1129-1141.

[17] Schönfeld, T. and Poinsot, T. Influence of boundary conditions in LES of premixed combustion instabilities. In *Annual Research Briefs of the Center for Turbulence Research, NASA Ames*, Stanford Univ, USA (1999):73-84.

[18] Moureau, V., Lartigue, G., Sommerer, Y., Angelberger, C., Colin, O. and Poinsot, T. Numerical methods for unsteady compressible multi-component reacting flows on fixed and moving grids. *J. Comput. Phys.* (2005) **202**(2):710-736.

[19] Mendez, S. and Nicoud, F. Large-eddy simulation of a bi-periodic turbulent flow with effusion. *J. Fluid Mech.* (2008) **598**:27-65.

[20] Roux, A., Gicquel, L. Y. M., Reichstadt, S., Bertier, N., Staffelbach, G., Vuillot, F. and Poinsot, T. Analysis of unsteady reacting flows and impact of chemistry description in large eddy simulations of side-dump ramjet combustors. *Combust. Flame* (2010) **157**:176-191.

[21] Haselbach, F. and Schiffer, P. Aiteb - an european research project on aero-thermodynamics of turbine endwalls and blades. *Int. J. of Therm. Sci.* (2007) **13**(2):97-108.

[22] Gomes, R. A. and Niehuis, R. Film cooling effectiveness measurements with periodic

unsteady iflow on highly loaded blades with main flow separation. In *Proceedings of ASME Turbo Expo 2009: Power for Sea, Land and Air* . Orlando, Florida, USA (2009).