EUROPEAN COMMISSION

Community research

is-enes
INFRASTRUCTURE FOR THE EUROPEAN NETWORK FOR EARTH SYSTEM MODELLING

IS-ENES2 MILESTONE (M -N°: 10.1)

Benchmark definition for evaluation of coupling strategies

CERFACS Working Notes, WN/CMGC/15/42

File name: IS-ENES2_M10.1_Benchmark_definition.docx or .pdf

Reporting period: 01/04/2013 – 30/09/2014

Author(s): Sophie Valcke

Release date: **13/10/2014**

Abstract

The first steps of the WP10 task 3 « Evaluation of coupling strategies » have been achieved as reported in this document. First, a series of mindmaps gathering the characteristics of the coupling technologies themselves but also the characteristics of the components and of the coupled model supported by the coupling technology was produced. Then coupling characteristics to benchmark in priority have been identified. Finally, a benchmark suite for evaluation of coupling strategies has been defined and is detailed here. It will consist of a number of pre-coded, stand-alone components running on different grids, a number of coupling configurations between these components, and a list of test cases that the different partners commit to implement.

Project co-funded by the European Commission's Seventh Framework Programme (FP7; 2007-2013) under the grant agreement n°312979		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants including the Commission Services	
RE	Restricted to a group specified by the partners of the **IS-ENES2** project	
CO	Confidential, only for partners of the IS-**ENES2** project	

Table of contents

Executive Summary

The milestone 10.1 (M10.1) “Definition of the benchmark suite for evaluation of coupling strategies » is linked to the WP10 task 3 « Evaluation of coupling strategies ». The first steps of this task have been successfully completed as reported in this document.

First, the possible functions/characteristics of coupled Earth System Models have been gathered in a series of mindmaps “CouplingTechnology”, “Components”, “Metadata”, “Composition” and “Deployment”. These mindmaps describe all the characteristics of the coupling technology itself and also the characteristics of the components and of the coupled model supported by the coupling technology.

Next the coupling characteristics to benchmark have been identified and prioritised. The priority characteristics that have been identified are the type of the component model grids, the number of MPI ranks used to run the component models, the number of fields exchanged between the components, and the frequency of exchange.

Finally, the benchmark suite for the evaluation of coupling strategies has been defined and is detailed here. The suite will consist of a number of pre-coded, stand-alone components running on different grids, a number of coupling configurations between these components, and a list of test cases that the different partners commit to implement.

1. Introduction

1.1 Objectives

The milestone 10.1 (M10.1) “Definition of the benchmark suite for evaluation of coupling strategies » is linked to the WP10 task 3 « Evaluation of coupling strategies ». The intention of this task is to define and implement a suite of coupled benchmarks based on simplified model components that capture the essence of the coupling challenges in European climate models without the complexities of the science. The first step in this task, which corresponds to M10.1, is to:

- capture the set of functional and performance characteristics that provide key constraints on coupling components of any Earth System Model (ESM) (see section 2);
- identify the priority coupling characteristics to benchmark (see section 3);
- define a suite of coupled benchmarks, i.e. a set of simplified components that will be coded with the aim of exercising the priority coupling characteristics (see section 4).

1.2 Context

The community work on the characterization of ESM coupling started during the « 2nd workshop on Coupling Technologies » held in Boulder in February 2013 during which a session "Coupling Technology Benchmarking" was organized. The participants were split into 3 groups who were asked to answer the following questions:

- What are the scientific and technical requirements, including functional (e.g., data exchange, regridding, etc.) and non-functional (e.g., performance, flexibility, etc.) aspects, to build a geophysical coupled system from independent models?
- What are the qualities that should be assessed in a coupling technologies benchmark and how should those qualities be measured? As a community, how can we progress in the realization of such a benchmark? What existing resources can we leverage to bootstrap the development of a community benchmark? ».

The notes of the 3 working groups were gathered into a document (see the Appendix). A few months later, the US project Earth System Bridge (https://earthsystemcog.org/projects/es-fdl/) proposed a first series of mindmaps gathering the different characteristics of existing coupling systems.

The IS-ENES2 WP10 T3 partners (MetO, UNIMAN, STFC and CERFACS) subsequently gathered in a 2-day workshop in Exeter on February 12-13, 2014. The first objective of the workshop was to complement the Earth System Bridge’s draft mindmaps with the material gathered during the Boulder Coupling Technology workshop and with any additional inputs prepared by the participants in order to produce a comprehensive characterization, or list of

possible functions/characteristics, for ESM coupling. This involved the participants to use the characterization and to identify a set of priority functions/characteristics for which well-targeted benchmark test cases were defined. This second task corresponds to the second objective of the current milestone.

2. Characteristics of ESM component coupling

The workshop produced a series of mindmaps gathering the possible functions/characteristics of coupled Earth System Models. The top level mindmap "CouplingSystem" references 5 sub mindmaps "CouplingTechnology", "Components", "Metadata", "Composition" and "Deployment" to describe all the characteristics of the coupling technology as well as the required characteristics of the components and of the coupled model supported by the coupling technology.

- The "CouplingTechnology" mindmap has 3 main branches: "architecture" to describe the basic design principles and other general characteristics of the technology (e.g. whether it offers fault tolerance, restart capability or dynamic load balancing), "implementation" giving more details about how the technology is implemented, and "utilities" that describes all the possible utilities offered by the technology such as a the use of graphical user interface or the production of code profiling;
- The "Components" mindmap describes the characteristics of the component models supported by the technology regarding their implementation language, their parallelism, their type of time step, the coupling data produced, etc;
- The "Metadata" mindmap details how the technology deals with metadata, whether it uses metadata as input for configuration, or whether it produces it, and what type of metadata it considers;
- The "Composition" mindmap describes what the technology offers in terms of the coupling model coupling data transformation and transport, the coupling modes supported (implicit, semi-implicit, explicit)
- The "Deployment" mindmap describes mainly the types of component mapping supported by the technology, i.e. how the component models can be mapped on the available computing resources.

After the workshop, additional discussions, also by teleconference with American colleagues, helped the mindmap to further evolve. For more details, the full set of mindmaps are available at https://github.com/IWCCT/es-fdl .

3. Priority coupling characteristics to benchmark

The benchmark suite will be targeted, in the first instance, towards the evaluation of the performance of the coupling with respect to the following characteristics identified from the comprehensive description of potential coupling characteristics offered by the mindmaps:

- Type of the component model grids
- Number of MPI ranks used to run the component models (up to $O(10^4)$)

- Numbers of fields exchanged between the components
- Frequency of exchange

Regarding the size of the coupling fields, the tests will be, in general, performed for 2 resolutions, i.e. for a typical "coarse" global grid of ~200 km resolution and a typical "high-resolution" global grid of 20-50 km. In addition, one test case will involve self-generated latitude-longitude grids thereby providing additional flexibility in resolution choices. This flexibility is expected to be useful for very high-resolution tests on the latest largest systems.

Code intrusion, development time and issues met during development and techniques for overcoming them, which are all aspects of 'ease of use', will also be evaluated and reported on, as they are very useful despite being difficult to quantify.

4. Definition of the coupled benchmark suite

To define a benchmark for the evaluation of coupling technologies, the following benchmark structure is proposed:

- A number of pre-coded, stand-alone components running on different types of grids
- A number of descriptions (configurations or specifications) of coupling between these components
- A list of test cases that will be implemented by the different partners

4.1 Stand-alone components

The stand-alone components will consist of simple, individual model codes containing no physics or dynamics but representative of real models in term of coupling characteristics. They will be coded in Fortran and will define a number of 2D Real Fortran arrays, which represent coupling fields that may be *required (in)* and *provided (out)* by the model when it is deployed in a coupled context. Each component will be implemented as a subroutine, or hierarchy of subroutines, wrapped into a driver and its coupling fields will appear as :

- IN and OUT arguments of the subroutine,
- arrays in shared modules,
- local data declared at a particular, possibly deep, level in the subroutine call tree.

The driver will enable a component to be executed in stand-alone mode. The simple case will be implemented where, on each time step, a component 'reads' with a set of required fields and 'writes' a set of provided fields. In the stand-alone mode, the coupling fields will be initialised from a utility library routine or a NetCDF file.

These components will internally be parallel MPI codes and will not refer to any coupling technology. They will use specific grids (see section 4.3) and will be parameterised in terms of the number of in/out fields and number of coupling exchanges in order to investigate the performance and scalability of different coupling technologies.

4.2 Coupling specifications and other benchmark characteristics

Coupling specifications describe how subsets of the stand-alone components are to be coupled in a particular benchmark test case, i.e. what will be their execution schedule and how the provided and required fields of the different components will be connected.

For each test case described in 4.3, coupling fields will be exchanged in both directions every time step and the run will cover a certain number of time steps, with both concurrent and sequential execution schedule of the two components.

In a concurrent schedule, a coupling field delivered by one model at the end of its coupling period is received by the other model at the beginning of the following coupling period and vice-versa. This ensures that both components run their time steps in parallel. The fields required by each component at the beginning of the first coupling period need to be "primed" by some means, e.g. read from a coupling restart file or assigned directly by the component. Concurrent coupling is also known as "asynchronous" coupling in the sense that for each coupling period, a component uses the fields produced by the other component during the previous period.

In a sequential schedule, one component uses some priming mechanism to get its input coupling fields at the beginning of the run. This component can then run for the number of time steps corresponding to the coupling period and provide its output coupling fields to a second component. The second component is then able to run. Meanwhile, the first component waits until the second component has run its coupling period. When the second component has completed is coupling period, the second component provides its output coupling fields to the first component, which is then able to run for a second coupling period. The second component then and waits, and so on. This schedule results in one component running when the other is waiting and vice-versa. We can note here that if the coupling exchanges occur every time step and if the work of the second component during its time step is null, the sequence of the first component sending a coupling field to the second followed by the second component sending back a coupling field to the first component is simply a "ping-pong" exchange.

The proposed timings to perform are:

- initialisation time,
- first time step,
- standard deviation, maximum, minimum and average over 99 additional time steps (so to give stable results)

The benchmark test cases will also propose mechanisms for checking correctness, as well as completion, of the coupled system. The test cases will also include basic instrumentation to provide metrics (such as timing and scalability).

4.3 Definition of test cases

The following stand-alone components running on the specified grids will be implemented by the different partners in the form described in section 4.1:

- Self-generated regular latitude-longitude grid allowing arbitrary resolutions to be used (*selfgenreglatlon*): STFC/UNIMAN
- Irregular, stretched and rotated latitude-longitude grid, following the ORCA configuration of the NEMO ocean model (*stretchlatlon*): CERFACS
- Quasi-uniform icosahedral mesh, following the atmospheric NICAM model (*icosa*): CERFACS
- Quasi-uniform cubed sphere mesh (*cubedsphere*) : MetO/UNIMAN

The partners will implement the coupling as defined in section 4.2 between the following components and will study the performance and scalability of the coupling with the specified technologies on the latest platforms available:

- *selfgenreglatlon* - *selfgenreglatlon* with same decomposition on both sides:
  - OASIS : CERFACS
  - OpenPALM : CERFACS
  - ESMF : STFC/UNIMAN
  - MCT : STFC/UNIMAN
- *selfgenreglatlon* - *selfgenreglatlon* with different decompositions on both sides:
  - OASIS : CERFACS
  - OpenPALM : CERFACS
  - ESMF : STFC/UNIMAN
  - MCT : STFC/UNIMAN
- *stretchlatlon* - *selfgenreglatlon*:
  - OASIS : CERFACS
  - OpenPALM : CERFACS
  - ESMF : MetOffice
- *icosa* - *icosa* :
  - OASIS : CERFACS
  - OpenPALM : CERFACS
  - ESMF : CERFACS
- *cubedsphere* (with finite differences) - *selfgenreglatlon* :
  - OASIS : MetO
  - ESMF : MetO
- *cubedsphere* (with finite elements) - *selfgenreglatlon*:
  - OASIS : MetO
  - ESMF : MetO

  Note : MetO/STFC/UNIMAN will develop a stand alone component but cannot commit to make a reference implementation ; this will depend on external developments in the GungHo and LFRic projects.

5. Perspectives

The mindmaps described in section 2 will be use by the US project Earth System Bridge as a starting point to develop a questionnaire to collect data about existing coupling technologies to be completed by the different coupling technology developers.

The test cases described in section 4 constitute the minimal set of cases the partners commit to implement. Additional features could be considered later if time permits; these include:

- Stand-alone components coded in languages other than Fortran
- 3D coupling fields
- I/O that can be considered as coupling between a component and a file
- More test cases implemented with OpenPALM

The resulting benchmark suite will be released to the community, including model developers, developers of coupling technologies and developers of computer systems. Experiments based on the benchmark suite will be realized by the WP partners on specific platforms. Results will be analysed and presented to the community, for example, at the future IS-ENES coupling workshop planned in April 2015 in Manchester, at the EGU and/or through publication in academic publications.

6. Appendix – Notes from the benchmark session of the 2nd workshop on Coupling Technologies, Boulder, Feb 2013

Functional Characteristics of a Coupling Technology

Data exchange and redistribution	
• Exchange between different models with different decompositions or within same model with same or different decompositions	• Yes/No • Types of grids supported • Masked grid supported (yes/no) • Types of decompositions supported • Local or global knowledge of decomposition • How easy to describe the decomposition • MOTR[1] for computation of communication patterns • MOTR for data exchange (for difficult test cases e.g. 40+ 3D fields every time step or load imbalanced cases)
• Exchange of data with halo	• Yes/No • Halo extension (one neighbor, more?)
• Support of adaptive grids	• Yes/No
Regridding and weight generation	
• Regridding of coupling data, sparse matrix multiplier	• Yes/No • Types of regridding supported • Dimensionality supported (2D, 3D, etc) • Types of grids supported • Types of decompositions supported • Externally generated weights supported • MOTR for regridding (performance) • Flexibility / ease of use •
• Regridding weight generation	• Yes/No • Types of regridding supported • Types of grids supported • Types of decompositions supported • Masked grids supported? • *MOTR for weight generation (performance)*
• Weight generation for non geometrical regridding (e.g. runoffs, catchment basins, calving)	• Yes/No • *MOTR for weight generation*
Mediation on coupling data	
• Averaging, accumulation	• Yes/No • *MOTR for operation*
• Combination of coupling fields	• Yes/NO
• Land-ocean coastline consistency checking	• Yes/NO

[1] MOTR: Measure of Time Required for

• Other types of mediation	• Yes/No, which ones
Time integration features	
• Implicit coupling	
• Calendar, clock	• Yes/No • Types of calendars • Length of time
Output handling	
• Coordinated stdout and stderr	• Yes/No • Human readable output?
• Online diagnostics	• Yes/No • Flexible, configurable?
• Output of state time	• Yes/No • Flexible, configurable?
• Diagnostics mediated with regridding, combinations, etc	• Yes/No • Regridding accuracy
Workflow support	
• Code extraction, build, run	• Yes/No • MOTR excluding runtime
• Configuration management	
• Cross model restart ability	
• Support for restart on system signal	
• Ensemble support including fault tolerance.	
• Coupled system launch support on complex parallel hardware	
• Archiving data	
• Case management	
Others	
• Encapsulation of different parts of the scientific code, separation of scientific knowledge from the “computer science” (e.g parallelism)	

Non-Functional Characteristics of a Coupling Technology

Portability and performance	
• Portability, standard compliance	• Number of compilers compiled with
• OS compatibility	• Number of platforms run on
• Multi-language support, language compatible	• Number of languages framework callable from

with ESM	
• Support for hybrid modes - openMP	• Yes/No
• Conformance to conventions (e.g. grid conventions)	• Yes/No, which ones:
• Memory impact , memory scalability of distributed data types	• Measure of memory impact
• Scalability	• Results of latency and bandwidth tests
• Low overhead	• Measure of overhead; compare different frameworks with same models
• Bit reproducibility	• Yes/No; details
• Fault tolerance	•
• Analysis tools	• Yes/No
• Dynamic load balancing	• Yes/No; how:
• Load balance measure	• *Measure performance of optimized result-- learn from HPC suppliers*
User support	
• Documentation / user guide	• Yes/No
• Human support by developers (mail, phone, etc)	• Yes/No • Average response time? quality
• Quality of prioritization process (feature requests)	•
• Community support	• Yes/No; how
• Code readability	• Poor/medium/high
Coupled model developer support	
• Range checking on input values/fluxes – outlier detection – validation checks	• Yes/No
• Profiling available	
• Debugging included, possibility to use with debuggers	• For the software • For the science
• Error handling	
• “Replay” mechanism to isolate a single component from feedbacks	
• Idealized test cases to check conservation, interpolation order	• Check mark exists? Tested?
• Test for numerical stability of coupled system	
User friendliness	
• Low intrusiveness, ease of use for legacy systems	• LOC changed • Time required to make model conformant (e.g. split into init.run/finalize) • Up-front work benefits vs benefits down the road
• Low entry point, shallow learning curve.	• Give same assignment to students and see if they do it right and how long it took and what they liked

• Ease of configuring runs, usability	•
• Availability of native data types, data types supported	•
• Possibility to change the version/resolution of the model	
• Community take-up	• *Number of groups, models*
Software development	
• Style and programming guides	• Yes/No
• Code quality, adherence to coding standards	• Measure of code quality (learn from commercial shops)
• Version control used?	• Yes/No
• Issue tracking used?	• Yes/No
• Unit test support	• Test coverage quality
• User extendable	• External developer check ins? LOC?
• Integration of user contributions, authority needed to modify code?	• Yes/No; method
• Stability of interfaces /data types/ maturity of the technology (i.e names shouldn't change)	•
• Longevity	• Is the group producing it likely to stay around • How many active developers
• Public access to the code	• Yes/No