EUROPEAN COMMISSION

Community research

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

IS-ENES2 DELIVERABLE (D -N°: 10.4)

Report on the suite of base benchmarks and analysis of performance on available platforms

{File name: IS-ENES2_D10.4.pdf}

Authors: ***G. Aloisio, C. Basu, J. Behrens, J Biercamp, A. Caubel, I. Fast, S. Fiore, M.-A. Foujols, P.G. Fogli, M. Hanke, S. Masina, S. Mocavero, P. Neumann, H. Struthers, S. Valcke***

Reviewers: ***P. Adamidis, M. Carter***

Reporting period: **01/04/2016 – 31/03/2017**

Release date for review: **24/03/2017** Final date of issue: **29/03/2017**

Revision table			
Version	**Date**	**Name**	**Comments**
1	24/03/17	Irina Fast	First release for internal review
2	29/03/17	Irina Fast	Final version with reviewers' comments integrated

Abstract

This report describes the final status of the ENES Benchmark Suite at the end of the IS-ENES2 project. The suite is assembled from applications of varying complexity ranging from computational and communication kernels to fully coupled Earth System Models (ESMs). The Redmine project management tool is employed to ensure the availability of up-to-date benchmarks and performance data. The HPC vendors' feedback suggests that the prepared benchmarks are useful and necessary for efficient collaboration between climate modelling community and HPC system providers.

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

List of figures

List of tables

List of Acronyms and Abbreviations

APE	Aqua Planet Experiment
CDO	Climate Data Operators
CESM	Community Earth System Model
CHSY	Core Hours per Simulated Year
CM	Climate Model
CMCC	Centro Euro-Mediterraneo sui Cambiamenti Climatici (Euro-Mediterranean Center on Climate Change)
CMIP5/6	Coupled Model Intercomparison Project Phase 5/6
CPMIP	Computational Performance Model Intercomparison Project
DKRZ	Deutsches Klimarechenzentrum (German Climate Computing Center)
DWD	Deutscher Wetterdienst (German National Meteorological Service)
ENES	European Network for Earth System modelling
ESM	Earth System Model
ESMF	Earth System Modeling Framework
FCA	Fabric Collective Accelerator
Flop	Floating point operation
FTP	File Transfer Protocol
GB	Gigabyte (10^9 bytes)
GFlop	10^9 Flop
HPC	High Performance Computer / Computing
ICON	Icosahedral Non-hydrostatic general circulation model
IDRIS	Institute for Development and Resources in Intensive Scientific Computing
I/O	Input / Output
IPCC AR5	Intergovernmental Panel on Climate Change Fifth Assessment Report
IPSL	Institut Pierre Simon Laplace
IS-ENES2	Infrastructure for the European Network of Earth System Modelling Phase 2
JRA	Joint Research Activity
LiU	Linköping University
LUCIA	Load balancing tool for OASIS coupled system
MCT	Model Coupling Toolkit
MD5	Message-Digest Algorithm 5
MPG	Max-Planck Institut für Meteorologie (in this document)
MPI	Message Passing Interface
MPMD	Multiple Program Multiple Data
MXM	Mellanox Messaging Accelerator
NEMO	Nucleus for European Modelling of the Ocean
NSC	National Supercomputer Centre at LiU
NWP	Numerical Weather Prediction
OASIS3-MCT	OASIS3 - Model Coupling Toolkit coupler

OpenPALM	Open source parallel coupler
RAM	Random Access Memory
RAPS	Real Applications on Parallel Systems
SYPD	Simulated Years Per Day
UEABS	Unified European Applications Benchmark Suite
UM	Uncoupled Model (in this document)
YAC	Yet Another Coupler

Executive Summary

This report describes the state of the ENES Benchmark Suite at the end of the IS-ENES2 project. Currently, the suite includes seven application benchmarks of different complexity and community coupling technologies benchmark. The available benchmarks are:

- CMCC-CM2 (ESM)
- EC-EARTH (ESM)
- IPSLCM (ESM)
- MPI-ESM1 (ESM)
- ICON (UM)
- Coupling technology benchmark based on OASIS3-MCT, OpenPALM, ESMF, MCT, and YAC
- NEMO tracer advection kernel
- ICON communication kernel

The coupled ESM benchmarks inherently reflect all requirements of Earth system modelling software on HPC systems and infrastructure. The availability of these benchmarks along with benchmarks for the evaluation of coupling technologies is a unique feature of the ENES benchmark suite distinguishing the ENES effort from the similar activities like RAPS or UEABS.

One of the main purposes of the collected benchmark suite is to strengthen the strategic partnership between the ENES consortium and HPC vendors. The benchmarks provide insights into computational characteristics of ESMs to HPC system providers and a basis for the targeted cooperation. Hence, the assembled benchmark suite was introduced to HPC vendors invited to the IS-ENES2 workshop series "Innovation in HPC for climate models", held in Hamburg in November 2016 – January 2017. The availability and access to the ENES benchmarks is generally appreciated by vendors as it helps them to gain familiarity with the codes outside of formal benchmark exercises associated with procurements. However, it was emphasized that the value of a benchmark could be lowered if it is too complex, hard to execute, or imposed high requirements on human resources and available benchmarking systems.

All available ENES benchmarks are listed on the ENES Portal (https://portal.enes.org/computing/performance/benchmarks) that provides information and services to the international Earth system modelling community and beyond.

1. ENES Benchmark Suite - Overview

The main objective of the JRA2 effort within the IS-ENES2 project is to assemble a set of benchmarks of varying complexity based on **real ESM codes** used in European **climate research** as well as to collect and make available key performance data on different HPC systems. The availability of such applications benchmarks is essential for more efficient collaboration between the climate modelling community and hardware/software vendors. In general, the prepared benchmarks can serve the following purposes:

- Benchmarking of the HPC systems for procurements
- Provide vendors a better way to assess performance characteristics of climate applications, thus fostering co-design and innovation
- Reduce the time needed for porting of climate research applications to newly procured systems
- Monitoring of system performance throughout the operational lifecycle, especially after machine, firmware, and software upgrades/updates
- Comparison of the performance of different ESMs on different computing systems in order to develop an understanding of factors affecting performance of ESMs
- Assessment of the porting or rewrite effort needed to run ESMs on new hardware architectures
- Provide testbeds to computer scientists for development and application of new algorithms and domain specific languages
- Support of European climate researchers in assessing the computing time requirements and selection of the most appropriate HPC system by providing performance references for different ESMs on different HPC systems

The core of the ENES benchmark suite is formed by four state-of-the-art coupled Earth System Models (ESMs). Furthermore, the new generation atmospheric model ICON, coupling technology benchmarks and performance relevant computational and communication kernels derived from NEMO and ICON models are included.

To ensure the availability and usability of the benchmarks and to allow for regular updates of benchmark codes and scripts, instructions for benchmark execution, and performance data we employ the Redmine project management tool. The top project "ENES Benchmark Suite" (https://redmine.dkrz.de/projects/enes-benchmark-suite) has a number of subprojects, each devoted to a single benchmark application or application category. The integration of such features as wiki, forums, issue trackers, and development roadmap made Redmine particularly appropriate for management of the suite. Furthermore, Redmine automatically tracks the number of downloads and computes MD5 checksums for provided files, thus supporting the dissemination of benchmarks.
The distribution of benchmarks underlies different licence agreements. In most cases the interested party has to approach the group providing the benchmark individually since the corresponding

licence agreements restrict the free distribution of these benchmarks. Contacts are provided for each benchmark if such a regulation applies.

In the next sections we provide descriptions of all benchmarks currently included into the ENES benchmark suite and outline future work after the end of the IS-ENES2 project.

2. ESM Benchmarks

The fully coupled ESM benchmarks represent the compute and data workloads that are characteristic for climate research applications and stress almost all features of a HPC system: floating-point performance, memory bandwidth, network interconnects, parallel filesystem performance etc. Four European state-of-the-art Earth system models form the core of the ENES benchmark suite. All four ESMs have contributed to the CMIP5 project, which have provided data for the IPCC AR5. Subsequent versions of these ESMs will participate in the CMIP6 project starting in 2017.
Below, detailed descriptions of the available ESM benchmarks and scalability measurements are provided. Computing throughput is measured in simulated year per days (SYPD) which is a standard measure for comparison of the computational performance of ESMs introduced by the CPMIP protocol [11].

2.1 CMCC-CM2 Benchmark

The CMCC–CM2 model [1] is the physical basis of the new CMCC (Centro Euro-Mediterraneo sui Cambiamenti Climatici) Earth System Model. The model is derived from the NCAR coupled model CESM, where the ocean component is NEMO [2] rather than the NCAR ocean model. In CMCC-CM2 all the climate components (atmosphere, ocean, land and sea-ice) are fully coupled via CPL7 CESM internal coupler.

Redmine project: https://redmine.dkrz.de/projects/cmcc-cesm-nemo-benchmark

Changes with respect to interim release (D10.2)

The ocean component of the CMCC-CM2 model has been updated to the latest stable release NEMO 3.6. A few bugs have been fixed and some work has been done in order to get better scientific results with respect to the version used for the previous benchmark.

Instructions on download, execution and analysis

IS-ENES2 partners can access the CMCC-CM2 model source code and input data through FTP. Permission can be requested by email to piergiuseppe.fogli@cmcc.it.

Documentation for the CESM and NEMO models can be found on their respective web site:

- CESM 1.2.2: http://www.cesm.ucar.edu/models/cesm1.2/
- NEMO 3.6: http://www.nemo-ocean.eu/About-NEMO/Reference-manuals/NEMO_book_3.6_STABLE

In particular the CESM User's Guide [3] explains in detail how to set up and run the CESM model. In the following we assume that the user is familiar with the CESM User's Guide and terminology (case, compset, etc.) and provide here a brief guidance through the steps required to run the CMCC-CM2 model:

1. Download model source code and data.
2. Set up the local input data directory.
   This directory contains input data required in order to run the model (initial and boundary conditions, interpolation weights, etc.) and should be created on a fast file system (e.g. GPFS or Lustre):

```
CESMDATAROOT=/path/to/data
export CESMDATAROOT
mkdir -p $CESMDATAROOT
cp cmcc_cm2_data.tar.gz $CESMDATAROOT
cd $CESMDATAROOT
tar xzf cmcc_cm2_data.tar.gz
```

   The `CESMDATAROOT` environment variable can be added to the user's shell configuration file.
3. Set up the model source code.

```
mkdir $HOME/CMCC-CM2
cp cmcc_cm2.tar.gz $HOME/CMCC-CM2
cd $HOME/CMCC-CM2
tar xzf cmcc_cm2.tar.gz
CCSMROOT=$HOME/CMCC-CM2/cesm
export CCSMROOT
```

4. Add the current platform to the list of supported machines.
   See Chapter 5 *"Porting and Validating CESM on a new platform"* of the CESM User's Guide. This step requires modifications of the files `config_compilers.xml`, `config_machines.xml` and the creation of `env_mach_specific.<platform>`, `mkbatch.<platform>` and `Depends.<platform>` in the directory `$CCSMROOT/scripts/ccsm_utils/Machines`. Use CMCC platform (athena) as a reference for NEMO specific settings.
5. Set up a new case which uses the NEMO model.
   This step sets up a new experiment with the global fully coupled model configuration (B compset) at ¼ degree horizontal resolution, which uses the NEMO ocean model. See section *"How to create a new case"* in Chapter 2 *"Creating and Setting Up A Case"* of the CESM User's Guide.

```
CASE=test01
CASEROOT=$HOME/CMCC-CM2/experiments/$CASE
cd $CCSMROOT/scripts
./create_newcase -case $CASEROOT \
-user_compset 2000_CAM5_CLM40%SP_CICE_NEMO_RTM_SGLC_SWAV \
-res f02_n0253 -mach <platform>
```

6. Configure the case.
   Set the processors layout following the section "*Changing the PE layout"*, in Chapter 2 "*Creating and Setting Up A Case"* of the CESM User's Guide.

```
cd $CASEROOT
# Modify file env_mach_pes.xml
emacs env_mach_pes.xml
./cesm_setup
```

   Once the total number of NEMO MPI tasks is chosen (`NTASKS_OCN`), the lat-lon task decomposition can be specified modifying the script `$CASEROOT/Buildconf/nemo.buildnml.csh` (variables `jpni`, `jpnj` and `jpnij`), adding the following lines to the namelist at section 6.1 of the script:

```
&nammpp
  jpni = NX
  jpnj = NY
  jpnij = ${NTASKS_OCN}
/
```

   where `NX*NY=NTASKS_OCN`. The user must update these values whenever the number of NEMO MPI tasks is changed.
   Note that NEMO does not support multithreading (OpenMP) so the variable `NTHRDS_OCN` in `env_mach_pes.xml` must always be 1.

7. Build the model.
   To build the model execute

```
./$CASE.build
```

   Note that during this step additional input data required by the model is automatically downloaded from the CESM input data repository and saved in the local input data directory `$CESMDATAROOT/inputdata`. See Chapter 3 *"Building CESM"* of the CESM User's Guide for further details.

8. Modify runtime settings.
   Modify runtime setting following section "*Customizing runtime settings*" in Chapter 4 "*Running CESM"* of the CESM User's Guide. On the CMCC computing platform (Athena) the best performance is obtained configuring the pio library to use the parallel NetCDF library for I/O:

```
./xmlchange -file env_run.xml -id PIO_TYPENAME -val pnetcdf
```

   Check for queue specific settings (wall clock time limits, project accounting, etc.) in the run script `${CASE}.run`.

9. Run the model.
   See section "*How do I run a case?*" in Chapter 4 "*Running CESM"* of the CESM User's Guide.

```
./$CASE.submit
```

10. Check for run successful termination
    Check for the string `SUCCESSFUL TERMINATION OF CPL7-CCSM` in the output from the job.
11. Timing analysis
    See section "*Load balancing a case*" in Chapter 4 "*Running CESM"* of the CESM User's Guide.

Performance reference

On the highest application level the performance of an ESM is related to two factors:

1. The performance of the individual component models (i.e. atmosphere, ocean etc.)
2. The load balance of the computational resources among all the components

In order to optimize the usage of available computational resources, it is important to load balance the component models in the best way trying to synchronize the execution and to minimize the processes idle time.
As in D10.2, the analysis of strong scalability of the CMCC-CM2 main components has been carried out, executing it in coupled mode. The main outcome of this analysis is the execution time of each component on a different number of nodes. This information allows us to define the best distribution of the computational resources, given the total number of cores allocated for the job.
The analysis of scalability has been performed on the ATHENA system, an iDataPlex cluster equipped with Intel Sandy Bridge cores, located at the CMCC Supercomputing Center. Details on the system configuration are summarized in the *Appendix A, Table A.1*.

The charts in *Figure 1*, *Figure 2* and *Figure 3* show respectively the execution time (for 5-day simulation), the simulated years per day (SYPD) and the speedup of the two main components: ocean and atmosphere. Reported analysis refers to the model executed at 1° resolution of in both components. The measurements for the NEMO model are displayed up to the limit of scalability since the coupled model performance is limited by the atmospheric model at this resolution.

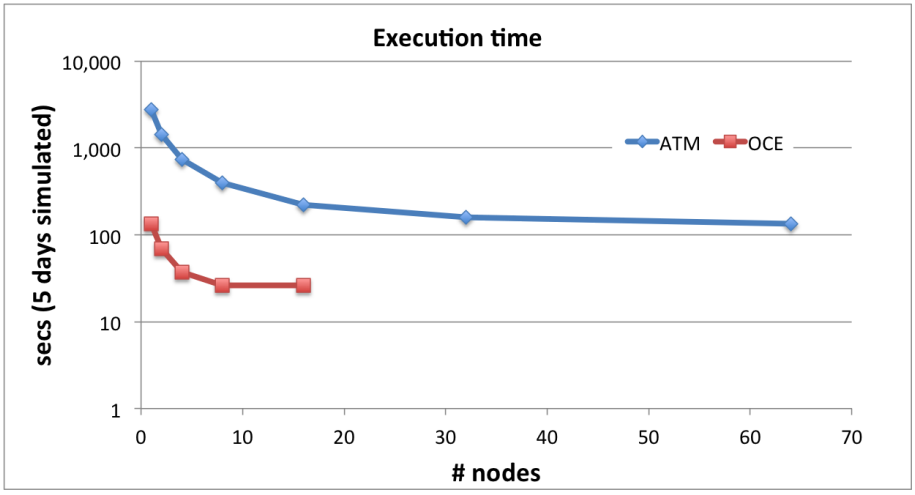

Figure 1: *Execution time of the main CMCC-CM2 components at 1°.*

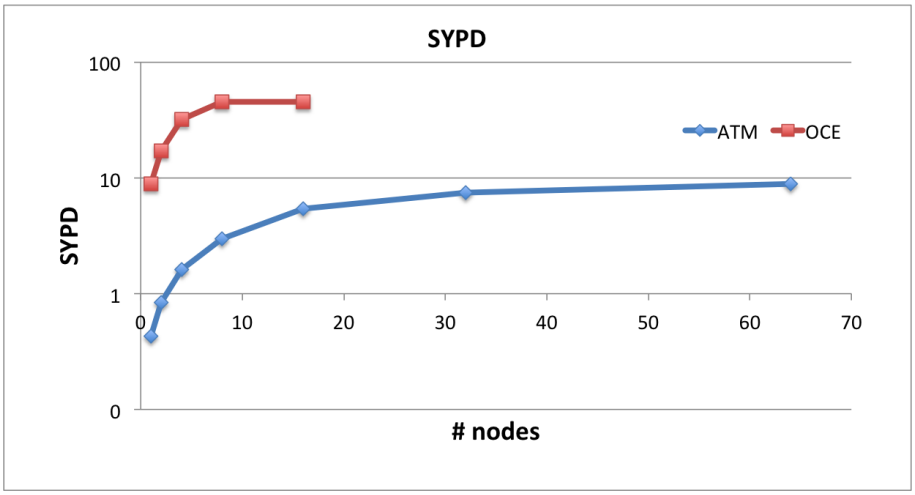

***Figure 2**: SYPD of the main CMCC-CM2 components at 1°.*

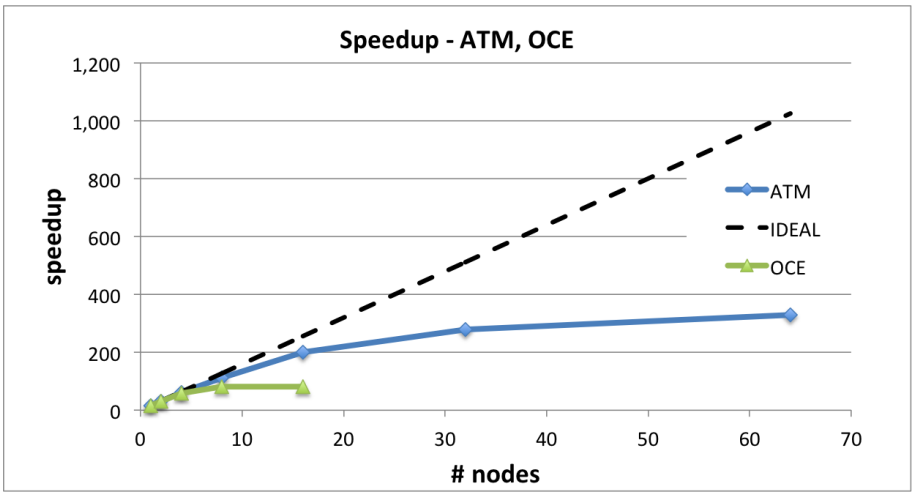

***Figure 3**: Speedup of the main CMCC-CM2 components at 1°.*

Performance analysis of the single components allows to define the best configuration (reported in *Figure 4*) on the Athena system, following the requirements of the CESM framework (fully concurrent except that the atmosphere runs sequentially with the ice, runoff, and land components; the coupler runs on a subset of the atmosphere processors, though concurrently with the land, ice, and runoff).

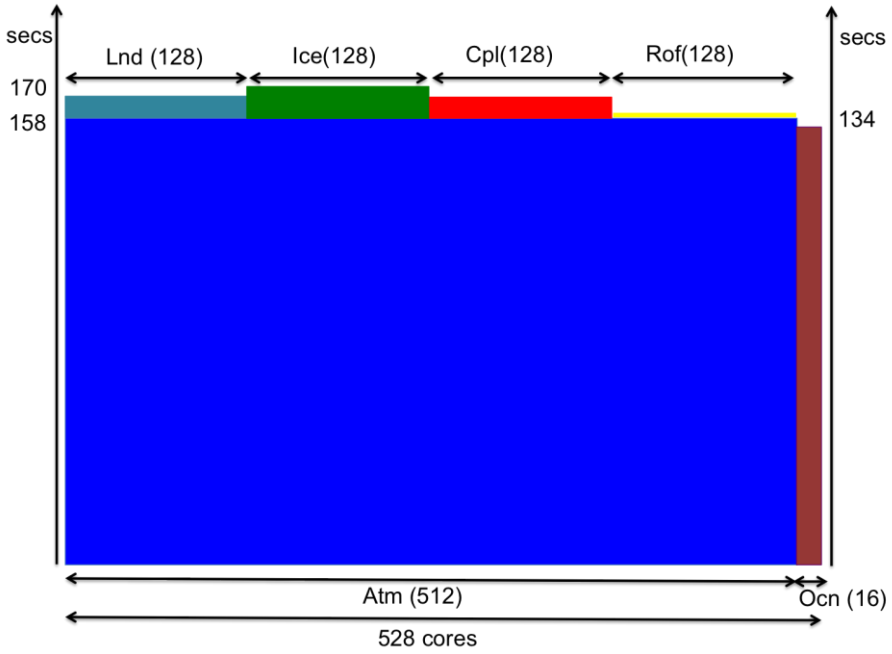

Figure 4: *CMCC-CM2 (at 1° resolution) best configuration on the Athena system.*

Table 1 reports the main metrics for ESM performance evaluation, their explanation and related values for the configuration represented in *Figure 4*.

Table 1: *CMCC-CM2 at 1° performance evaluation metrics according to CPMIP definition [11].*

Performance metric	Explanation	Value
Resolution	Grid point distance	Atm, Lnd: 0.9 x 1.25, 100km (lat) x 100km (lon) (L30) Ocn, Ice: ~100Km (L50) Rof: ~56Km (0.5°)
Complexity	Number and dimension of variables (2D or 3D) in restart files of the main components (Atm and Ocn)	Atm (2D): 200 Atm (3D): 126 Ocn (2D): 20 Ocn (3D): 23
SYPD	Simulated years per day	~ 7.69
CHSY	Core hours per simulated year	~ 1648
ASYPD	Actual simulated years per day (taking into account queue wait time)	~ 6.8
Memory bloat	Actual and ideal memory consumption	Actual (Ocn+Atm): 7 GB Ideal (Ocn+Atm): 2.5 GB
Coupler cost	Time spent in coupling/overall time	4.7%
Load imbalance	Time spent waiting for coupler (or other components) / overall time	3.3%
#Grid points		Atm: ~ 1.658.880 Ocn: ~ 5.285.200
Parallelisation	Resources allocation allowing the best load balance among components	512 (atm) + 128 (ice) + 16 (ocn) + 128 (lnd) + 128 (rof) + 128 (cpl); lnd, ice, cpl, rof sequential to atm

The same analysis has been performed on the model at ¼° resolution for 1-day simulation. *Figure 5* reports the best configuration. Ocean component is executed on the minimum number of cores which satisfies the memory requirement. The values for performance metrics are reported in *Table 2*.

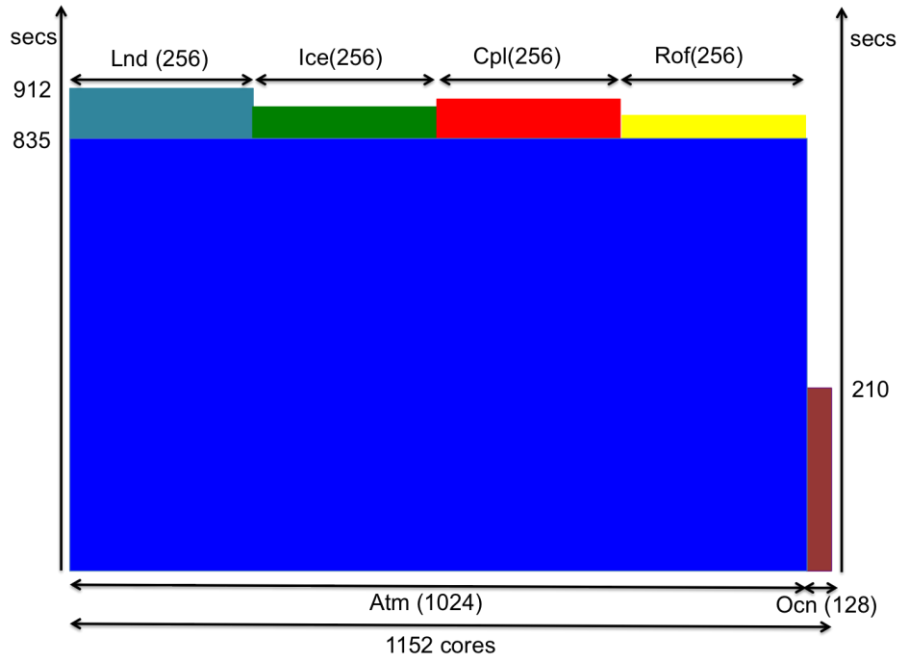

Figure 5: *CMCC-CM2 (at 1/4° resolution) best configuration on the Athena system.*

Table 2: *CMCC-CM2 at 1/4° performance evaluation metrics according to CPMIP definition [11].*

Performance metric	Explanation	Value
Resolution	Grid point distance	Atm, Lnd: 0.23 x 0.31, ~26km (lat) x ~35km (lon) (0.25L30) Ocn, Ice: ~28Km (0.25L50) Rof: ~56Km (0.5°)
Complexity	Number and dimension of variables (2D or 3D) in restart files of the main components (Atm and Ocn)	Atm (2D): 200 Atm (3D): 126 Ocn (2D): 26 Ocn (3D): 23
SYPD	Simulated years per day	~ 0.23
CHSY	Core hours per simulated year	~ 106,839
ASYPD	Actual simulated years per day (taking into account queue wait time)	~ 0.18
Memory bloat	Actual and ideal memory consumption	Actual (Ocn+Atm): 500 GB Ideal (Ocn+Atm): 39 GB
Coupler cost	Time spent in coupling/overall time	3.0%
Load imbalance	Time spent waiting for coupler (or other components) / overall time	11.9%
#Grid points		Atm: ~ 26.542.080 Ocn: ~ 75.777.100
Parallelisation	Resources allocation allowing the best load balance among components	1024 (atm) + 256 (ice) + 128 (ocn) + 256 (lnd) + 256 (rof) + 256 (cpl); lnd, ice, cpl, rof sequential to atm

2.2 EC-Earth Benchmark

The Earth System Model EC-Earth [4] is developed as part of a Europe-wide consortium. The components of the EC-Earth model are IFS for the atmosphere, NEMO for the ocean, LIM for the sea-ice, and HTESSEL for the land surface and vegetation coupled through the OASIS coupler.

Redmine project: https://redmine.dkrz.de/projects/ec-earth-benchmark/

Changes with respect to interim release (D10.2)

The model repository branch `/ecearth3/branches/tuning/3.2beta/main` is used for benchmarking. It is a pre-release of the CMIP6 version of the EC-Earth model.

Instructions on download, execution and analysis

For license agreement reasons the access to the EC-Earth benchmark must be requested individually by email to ralf.doescher@smhi.se. EC-Earth development portal wiki (https://dev.ec-earth.org/) provides information on compilation, porting and running of the EC-Earth model.

Performance reference

The benchmark specifics are:

- Coupled IFS+NEMO, launched using the ece-ifs+nemo.sh EC-Earth 3.2 run script
- IFS (T255L91), NEMO (ORCA1L75_LIM3)
- Three month simulations starting 1990-01-01
- Coupling frequency: 2700 sec
- IFS time step: 2700 sec
- NEMO time step: 2700 sec
- LIM3 time step: 2700 sec

The measurements were performed on the Beskow HPC system (see *Appendix A, Table A.2*) using Intel and Cray compiler with the following options:

1. CRAY compiler
   - General Fortran flags for compiling (note a small number of subroutines compiled with `-O0`)
     ```
     -sreal64 -em -hnoomp -O2
     ```
   - General C flags for compiling
     ```
     -O3
     ```
   - Preprocessor macros for IFS source code
     ```
     linux LINUX LITTLE LITTLE_ENDIAN POINTER_64 BLAS
     ```
   - NEMO Fortran flags
     ```
     -em -s integer32 -s real64 -O2 -e0 -eZ
     ```
2. Intel compiler
   - General Fortran flags for compiling
     ```
     -O2 -fp-model precise -xHost -g -traceback -r
     ```

- General C flags for compiling

```
-O2 -g -traceback -xHost
```

- Preprocessor macros for IFS source code

```
linux LINUX LITTLE LITTLE_ENDIAN POINTER_64 BLAS
```

Figure 6 shows results of the load balancing analysis for the EC-Earth model performed with the LUCIA tool [6]. The atmospheric component IFS has been run on 288 cores, the ocean component NEMO (together with LIM) on 96 cores, runoff mapper on 1 core, and I/O servers (XIOS) on 1 core. The runtime of the binary generated with the Intel compiler is 30 min 13 sec which corresponds to 11.75 SYPD. With the Cray compiler the measured runtime is 26 min 39 sec which corresponds to 13.32 SYPD.

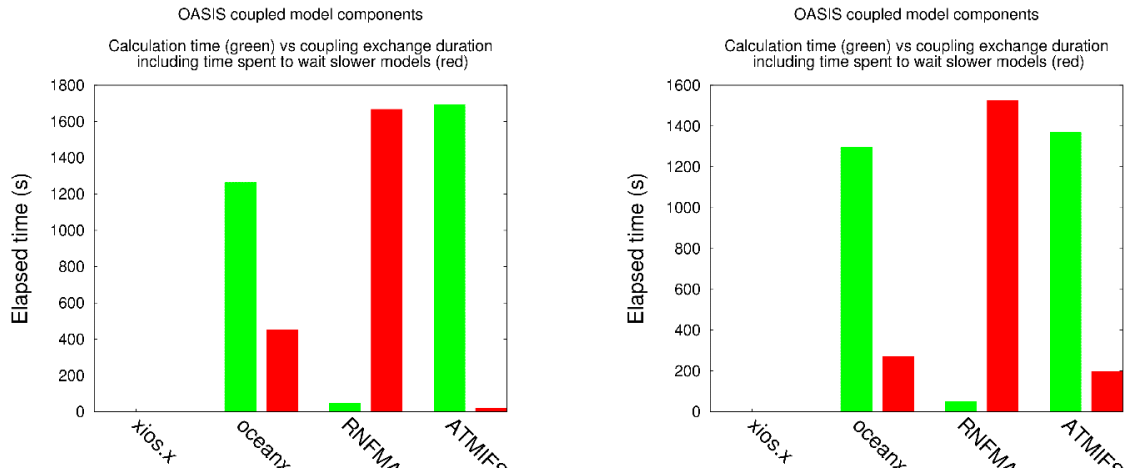

Figure 6: *LUCIA load balancing analysis for EC-EARTH model compiled with Intel compiler (left) and Cray compiler (right).*

The performance of the individual component models IFS and NEMO has been analysed in depth using the Allinea Performance Reports (http://www.allinea.com/products/allinea-performance-reports/) tool which gives an insight into the percentage of time spent in compute, MPI and I/O parts of the code. The I/O part includes time spent in MPI-IO calls and system library calls (e.g. `read`, `write`, and `close`) needed to read input data from and write output data to the filesystem.
Table 3 and *Figure 7* show scaling results for the IFS model.

Table 3: *Scaling characteristics of the IFS model compiled with Intel and Cray compiler.*

	Intel compiler		Cray compiler	
#Nodes (cores)	**run time**	**SYPD**	**run time**	**SYPD**
5 (160)	42 min 21 sec	8.38	35 min 21 sec	10.04
10 (320)	27 min 31 sec	12.90	22 min 21 sec	15.89
15 (480)	22 min 14 sec	15.97	19 min 28 sec	18.24
20 (640)	20 min 33 sec	17.28	17 min 25 sec	20.39
30 (960)	19 min 14 sec	18.46	16 min 44 sec	21.22

Figure 7: *Scaling characteristics of the IFS model comparing Intel and Cray compiler.*

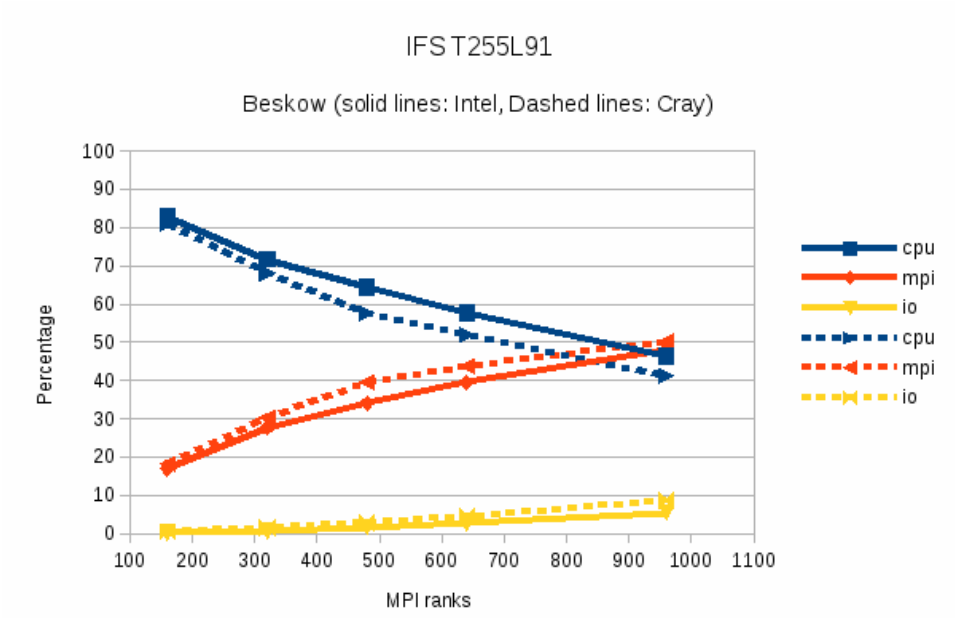

Figure 8: *Allinea Performance Reports results for the atmospheric component IFS of the EC-EARTH model. Proportion of time spent in compute, MPI and I/O parts is shown.*

The NEMO standalone configuration has been analysed without XIOS since Allinea Performance Reports cannot be used for MPMD applications yet. The scaling characteristics of the NEMO model are presented in *Table 4* and *Figure 9*. Proportion of the execution time spent in compute, MPI and I/O parts of NEMO for different number of cores is shown in *Figure 10*.

Table 4: *Scaling characteristics of the NEMO model compiled with Intel and Cray compiler.*

	Intel compiler		Cray compiler	
#Nodes (cores)	**run time**	**SYPD**	**run time**	**SYPD**
2 (64)	35 min 11 sec	10.09	34 min 11 sec	10.39
3 (96)	24 min 44 sec	14.36	25 min 03 sec	14.17
4 (128)	20 min 08 sec	17.64	21 min 08 sec	16.80
6 (192)	16 min 15 sec	21.85	17 min 19 sec	20.50
8 (256)	13 min 43 sec	25.89	15 min 42 sec	22.61
12 (384)	12 min 48 sec	27.74	16 min 53 sec	21.03

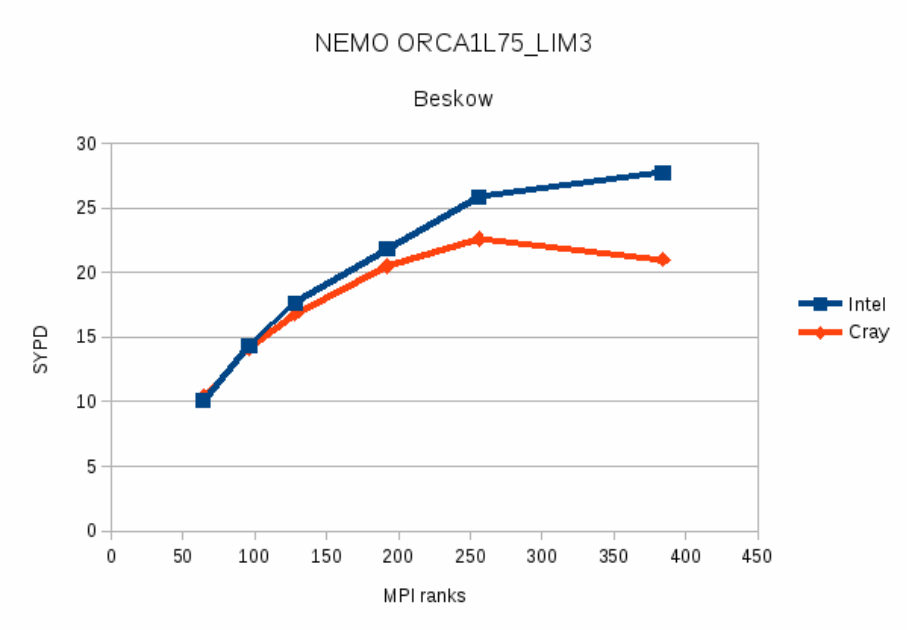

Figure 9: *Scaling characteristics of the NEMO model comparing Intel and Cray compiler.*

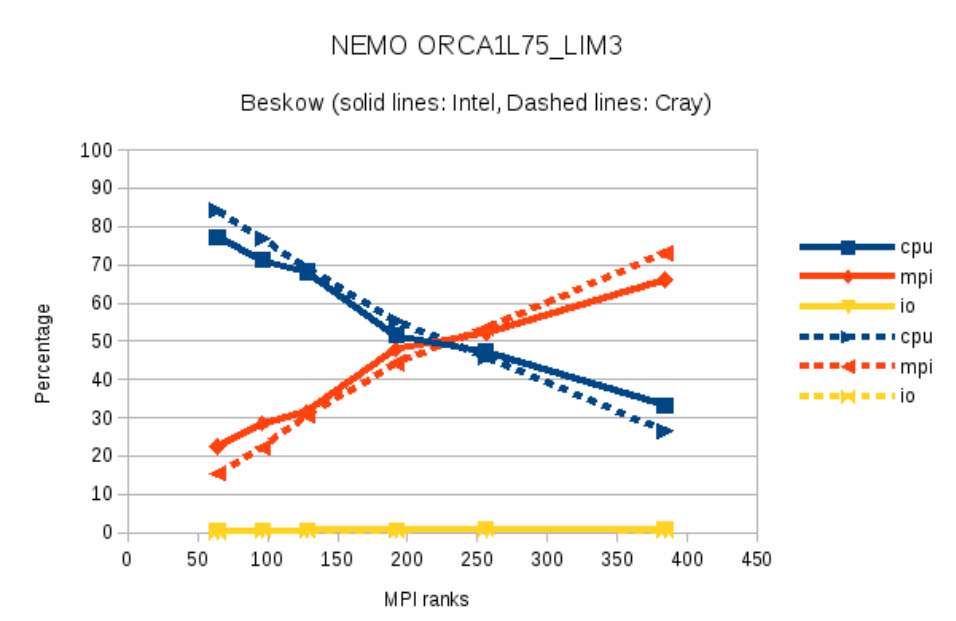

Figure 10: *Allinea Performance Reports results for the ocean component NEMO of the EC-EARTH model. Proportion of time spent in compute, MPI and I/O parts is shown.*

2.3 IPSLCM Benchmark

IPSL coupled Earth system model [5] is a full ESM. In addition to the physical atmosphere-land-ocean-sea ice model, it also includes a representation of the carbon cycle, the stratospheric chemistry and the tropospheric chemistry with aerosols. The IPSLCM benchmark is available for two different generations of the model: IPSLCM5A and IPSLCM6. The IPSLCM6 model includes LMDZ as atmospheric model, NEMO as ocean model, LIM2/LIM3 as sea ice model, PISCES as marine biogeochemistry model, ORCHIDEE as land model, and INCA as chemistry and aerosol model (*Figure 11*). Ocean and atmospheric components are coupled via OASIS3-MCT parallel coupler. XIOS I/O library is used to manage the model output.

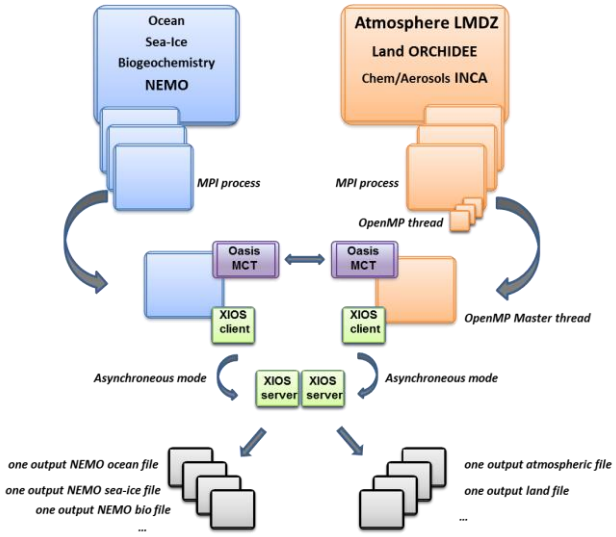

***Figure 11**: Components of IPSLCM6 model.*

IPSLCM5A and IPSLCM6 benchmarks are configured to simulate a short time period (3 months) and are automatically launched for monitoring on two different supercomputers: Curie (Bull cluster at TGCC, see *Appendix A, Table A.*3) and Ada (IBM xSeries cluster at IDRIS, see *Appendix A, Table A.4*). IPSLCM5A is the version that was employed within the CMIP5 project. It has been used for scientific benchmarking and regression testing (*trusting*) since 2010. IPSLCM6 is currently under development through an agile method with more than 12 versions being produced over the last 18 months with two model resolutions: VLR (very low resolution) and LR (low resolution). The IPSLCM6-VLR is used for regular testing (every 2 days) and IPSLCM6-LR is used for testing once per week. A variety of quality control tests with respect to reproducibility, restartability, and reliability has been defined. Summarized information on regular trusting results is available through a web service at http://webservices.ipsl.jussieu.fr/trusting/.

Redmine project: https://redmine.dkrz.de/projects/ipsl-cm/

Changes with respect to interim release (D10.2)

The IPSLCM5A benchmark is frozen. The IPSLCM6 benchmark is not yet released since the CMIP6 version is still under development.

Instructions on download, execution and analysis

The benchmark includes source code, utilities and additional files required to run and check a short simulation period of 3 months. IPSLCM5A benchmark (a tar file including: source code, compiling tools, input files, output files as examples, and `README`) is available on request by email to Arnaud.Caubel@lsce.ipsl.fr. All instructions on build and execution procedure can be found in the included `README` file.

Performance reference

IPSLCM is based on three different executables running simultaneously. Therefore, the load balancing requires particular care. The release candidate 0 of the IPSLCM6 model has been used to determine the number of cores required by each executable to keep a balanced workload for two different resolutions. For IPSLCM6-VLR with 96x95x39 atmosphere resolution mesh (LMDZ) and 182x149x31 ocean resolution mesh (NEMO ORCA2), 27 simulated years per day could be achieved with a total of 128 cores. For IPSLCM6-LR with 144x142x79 atmosphere resolution mesh (LMDZ) and 362x332x75 ocean resolution mesh (NEMO eORCA1), 6 simulated years per day could be achieved with a total of 480 cores.

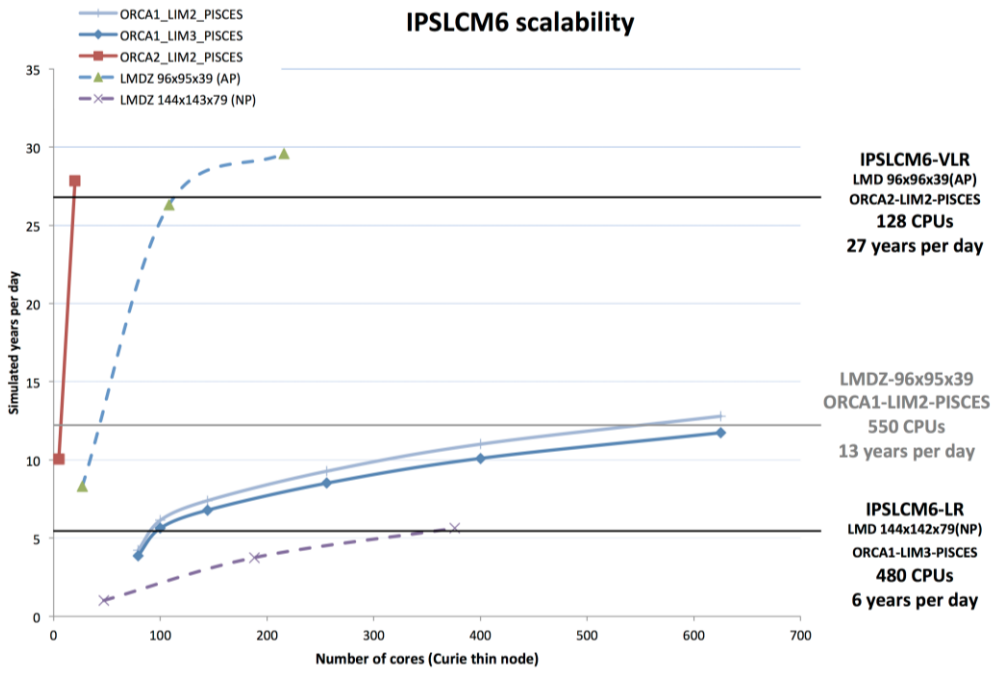

Figure 12: *IPSLCM6 scalability. The figure shows the number of simulated years per day for atmospheric and oceanic component of IPSLCM at different resolution and on different numbers of cores on Curie, TGCC. For IPSLCM6-VLR with 96x95x39 atmosphere resolution mesh (LMDZ) and 182x149x31 oceanic resolution mesh (NEMO ORCA2 LIM2 PISCES), 27 simulated years per day could be achieved with a total of 128 cores. For IPSLCM6-LR with 144x142x79 atmosphere resolution mesh (LMDZ) and 362x332x75 oceanic resolution mesh (NEMO eORCA1 LIM3 PISCES), 6 simulated years per day could be achieved with a total of 480 cores. For the third possible configuration with 96x95x39 atmosphere resolution mesh (LMDZ) and 362x332x75 oceanic resolution mesh (NEMO eORCA1 LIM2 PISCES), the number of cores used by the ocean could be increased and 13 simulated years per day could be achieved with a total of 550 cores.*

Figure 12 shows that the maximum speed of the whole coupled system will be approximately the same as the "slowest" component. Consequently, the optimization work consists in finding the optimal scalability for the slowest component and attributing resources for each model according to this number. This work was done using LUCIA [6], a load balancing tool for Oasis coupled systems.

2.4 MPI-ESM1 Benchmark

MPI-ESM1 (Max-Planck Institute Earth System Model 1) [7] is a state-of-the-art global Earth System Model consisting of components ECHAM6/JSBACH that includes atmosphere, land surface, soil, and vegetation processes and MPIOM/HAMOCC that includes ocean, sea ice and ocean biogeochemistry. ECHAM6/JSBACH and MPIOM/HAMMOC run as separate executables coupled via the OASIS3-MCT coupler.

Redmine project: https://redmine.dkrz.de/projects/mpi-esm-benchmark

Changes with respect to interim release (D10.2)

None. The next release will be provided as soon as the final CMIP6 version of MPI-ESM1 is available.

Instructions on download, execution and analysis

All public releases of the MPI-ESM1 benchmark package are stored for download on the Files area of the Redmine project "MPI-ESM1 Benchmark". The corresponding input data sets can be downloaded from the Cloud Storage maintained by DKRZ as specified in the Redmine project. The benchmark package includes source code files, build, run and evaluation scripts. General guidelines for compiling, running, and verifying numerical correctness of the results are provided on the Redmine Wiki. Up-to-date instructions can be found in the README.txt file distributed with the benchmark package.

Performance reference

The benchmark test case simulates the historical climate in years 1850-1851 at MR spatial model resolution. The benchmark was executed on the DKRZ Mistral phase 1 system (see *Appendix A, Table A.5* for system configuration) using

- Intel Compiler version 16.0.1
- Bullx MPI version 1.2.8.3 with Mellanox libraries MXM (version 3.3.3002) and FCA (version 2.5.2393)
- CDO version 1.7.0 (for evaluation of correctness of benchmark execution)

The time measurements have been performed using internal model timers.

Figure 13 shows the scalability curve for MPI-ESM1. The measurements cover the processor range from 24 cores (1 Mistral node) to 864 cores (36 Mistral nodes), which corresponds to an ideal (linear) speedup by a factor of 36. The actual speedup amounts to 16. The scalability of the coupled model is limited by the behaviour of the atmospheric component ECHAM. According to the detailed

profiling analysis the scalability of ECHAM is affected by non-scaling global data transpositions needed to re-arrange data between spectral and grid point spaces. The other performance bottleneck is high-frequency (every 6 simulated hours) serial data output: the percentage of wall-clock time spent for output increases from 5% on 24 cores to 57% on 864 cores (not shown here). This issue is resolved in the ECHAM6.3 version by implementing asynchronous parallel output via dedicated I/O servers (CDI-pio).

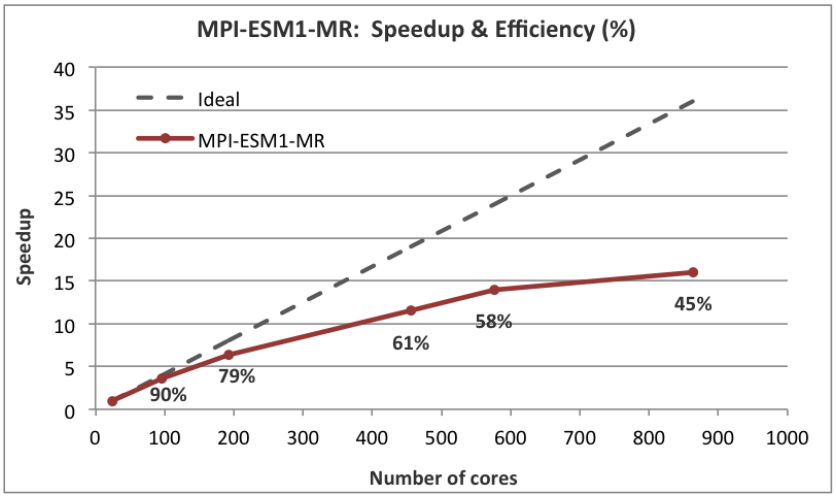

Figure 13: *Linear (idealised) and measured speedup curves for MPI-ESM1 model. Numbers at data points indicate the parallel efficiency defined here as ratio between actual speedup and ideal speedup. Note that speedup and efficiency data refer to the execution time on 24 cores.*

The measured benchmark execution times correspond to the number of simulated years per day shown in *Figure 14*. Throughput rates above 20 SYPD (which is a prerequisite for spin up, tuning and performing of ensemble experiments within reasonable time frames) can be achieved for total number of cores higher than 576 (24 Mistral nodes).

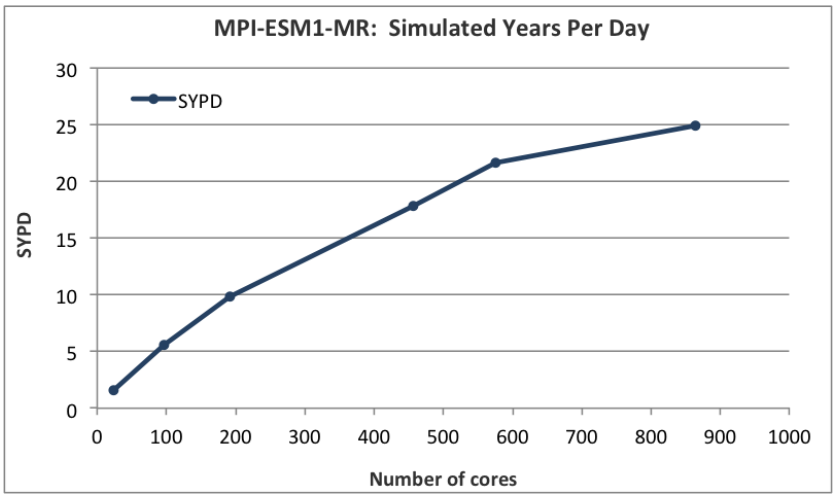

Figure 14: *Simulated Years per Day that can be achieved with MPI-ESM1-MR using different numbers of cores.*

3. Uncoupled model benchmarks

Obviously, the computational throughput of a coupled ESM is limited by the performance of the slowest component model. Therefore, prior analysis of uncoupled model components is necessary since this prevents misleading interpretation of performance data due to load imbalance of components. The uncoupled models also lend themselves for deeper performance analysis because many tools exhibit problems when examining MPMD applications.

3.1 ICON Benchmark

ICON (ICOsahedral Non-hydrostatic model) [8] is the new generation unified weather forecasting and climate model jointly developed by Max-Planck-Institut für Meteorologie (MPG) and Deutscher Wetterdienst (DWD). ICON employs triangular and hexa-/pentagonal grids arising from iterative subdivision of the edges of an icosahedron (polyhedron with 20 triangular faces) which is mapped onto the globe. The uncoupled ICON benchmark uses the atmospheric component of the ICON model only. The dynamical core of the model solves the fully compressible non-hydrostatic equations of motion.

Redmine project: https://redmine.dkrz.de/projects/icon-benchmark

Changes with respect to interim release (D10.2)

The last released ICON benchmark, version 16.0, has been designed in close collaboration with the centre of excellence ESiWACE and executes an aqua-planet experiment (APE) using various resolutions (results provided in benchmark for global resolutions up to 5km; results obtained at DKRZ for global resolutions up to 1km). The benchmark has been completely redefined compared to the previous release 13.0 to account for the recent developments of ICON physics.

Instructions on download, execution and analysis

All public releases of the ICON Benchmark are provided for download on the Files tab in the Redmine sub-project "ICON download area". The download area for the ICON benchmark is accessible for registered and approved users only. The registration procedure is described in the publically accessible Redmine project "ICON Benchmark". Input data sets are provided on DKRZ Cloud Storage which can be reached through a link from the Redmine project.

The ICON benchmark package includes source code files, build, run and evaluation scripts. General guidelines for compiling, running, and evaluation of the benchmark are provided on Wiki in the Redmine project "ICON Benchmark". Up-to-date instructions can be found in the `README_exp.APE_benchmark` file distributed as part of the ICON benchmark package.

Performance reference

The ICON APE benchmark contains a number of predefined test cases (corresponding to different model resolutions) as summarized in the *Table 5*:

Table 5: *Test cases provided with ICON APE benchmark.*

Model Grid	R2B4	R2B5	R2B6	R2B7	R2B8	R2B9
Horizontal resolution	140 km	70 km	35 km	18 km	8 km	5 km
Number of vertical levels	90	90	90	90	90	90
Number of grid cells	20480 x 90	81920 x 90	327680 x 90	1310720 x 90	5242880 x 90	20971520 x 90
Time step	600 s	300 s	120 s	60 s	30 s	15 s

The simulated time period is set to 200 time steps for all resolutions (if the focus is on comparison of different resolutions then a conversion to SYPD is recommended). The model output is activated. The benchmark was executed on the phase 2 part of the DKRZ cluster Mistral (described in *Appendix A, Table A.5*), using

- Intel Compiler version 17.0.1
- Bullx MPI version 1.2.8.3 with Mellanox libraries MXM (version 3.3.3002) and FCA (version 2.5.2393)

Figure 15 and Figure 16 show the speedup curves for ICON APE, R2B5 and R2B9 resolutions. At the end of the respective speedup curve the number of grid points per MPI task is a factor 64 higher for the R2B9 test case compared to the low resolution R2B5 test case. Therefore, the high resolution test case still shows a near ideal behaviour while the other does not. Computational resources available on Mistral do not allow us to extend the R2B9 measurements up to the scalability limit.

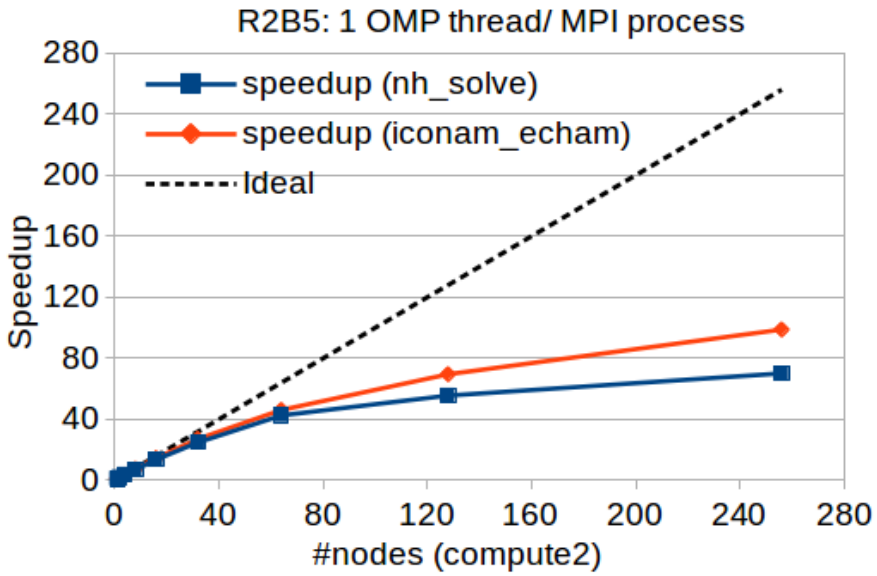

Figure 15: *Linear and measured speedup curves for ICON APE benchmark using a R2B5 grid with 90 vertical levels.*

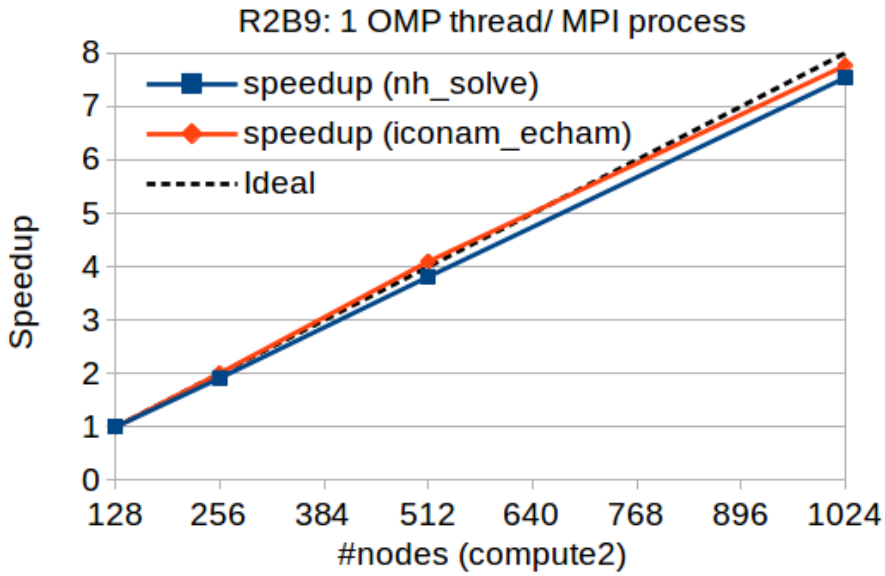

Figure 16: *Linear and measured speedup curves for ICON APE benchmark using a R2B9 grid with 90 vertical levels.*

The intra-node as well as the node-level performance data are provided in the performance section of the "ICON Benchmark" Redmine project.

4. Coupling technology benchmarks

On the road to the extreme scale computing, the coupling performance is becoming a major challenge in NWP and ES models. Different coupling technologies are used today to link ESM components (e.g. the atmosphere, oceans, land, sea ice, etc.) and although their implementations differ vastly, they typically carry out similar functions such as managing data transfer between two or more components, interpolating the coupling data between different grids, and coordinating the execution of the constituents.

Within the IS-ENES2 project five coupled configurations have been made available for testing in a standard benchmarking environment using "toy" models to allow us to focus on the coupler performance. The components of these test cases run on regular latitude-longitude grids with 1000x1000 and 3000x3000 grid points. They exchange couplings fields back and forth using each of the following couplers:

- OASIS3-MCT
- OpenPALM
- ESMF
- MCT
- YAC

A detailed description and evaluation of coupler benchmarks is covered in a separate IS-ENES2 deliverable D10.3 which can be downloaded from the IS-ENES2 web page (https://portal.enes.org/ISENES2/documents/deliverables/). The benchmarks are available upon request at Sophie.Valcke@cerfacs.fr.

5. Kernel benchmarks

Kernels are complementary to full application benchmarks. The focus of a kernel lies on specific parts of the model and helps to understand, analyse and improve these parts. The increased efficiency of working with kernels results directly from reduced and less complex source code, build process, execution effort and software environment requirements.
However, kernel performance measurements do not necessarily give a realistic estimate of the performance of the same part within the full model. This is due to different workloads, cache utilization and shifted process timelines (e.g. collective communication completion depends strongly on how process activities are shifted in time relative to each other). On the other hand, kernel performance measurements can disclose important runtime properties that can be difficult to identify within the workload noise of the full model.
Below we describe the tracer advection kernel derived from the widely used ocean model NEMO and a communication kernel derived from the new generation atmospheric model ICON.
Furthermore, kernels representing compute-intensive algorithms and communication patterns that are characteristic for climate models will be part of the suite.

5.1 NEMO Tracer Advection Kernel

Starting from the NEMO code profiling results, one of the most computational intensive routine is the `tra_adv_muscl`, which implements the Monotone Upstream Scheme for Conservative Laws (MUSCL) for tracer advection [9]. In the MUSCL formulation, each tracer is evaluated at velocity points assuming a linear tracer variation between two adjacent T-points. The implementation follows these steps:

- Horizontal advective fluxes
  - First guess of the slopes
  - Boundaries exchange among processes
  - Evaluation of the slopes
  - Evaluation of the MUSCL fluxes
  - Boundaries exchange among processes
- Vertical advective fluxes
  - Evaluation of the slopes
  - Evaluation of the MUSCL fluxes

The implementation uses three nested loops along the three spatial dimensions and the MPI communication uses a cross pattern (with point-to-point calls). It was selected since its basic code structure and the operations involved are representative of the whole code.

Redmine project: https://redmine.dkrz.de/projects/nemo-kernels

Changes with respect to interim release (MS10.3)

Minor revision of the array initialisation. Introduction of an internal timer using `MPT_Wtime` and of more repetitions (with the number of iterations set to 100) to increase the significance and robustness of the time measurements.

Instructions on download, execution and analysis

Users can download two files from the Files tab of the Redmine project "NEMO Kernels":

1. The kernel code (`tra_adv.F90` or `tra_adv_iter.F90`) implements the MUSCL scheme, which can be compiled and run without linking external libraries
2. An example of the run script (`run_job.sh`) to execute the tests. The script refers to the execution on the CMCC HPC system described in the *Appendix A, Table A.1*. It includes directives for the LSF batch system and instructions for the submission using LSF

The setup defines two test domains: the first one includes about $16x10^6$ grid-points and it is similar to the Mediterranean Forecast System (MFS) configuration used at CMCC; the second domain (Big, approx. $265x10^6$ grid points) is defined in order to saturate the available memory on the compute node. Users can change the horizontal grid acting on `(ljpi_mfs, ljpj_mfs)` and `(ljpi_big, ljpj_big)` respectively for the two test domains, while the number of vertical levels can be modified by using the `JPK` variable.

Performance reference

The performance results reported here are related to the Athena HPC system described in the *Appendix A, Table A.1*.

Table 6 and *Table 7* report the execution time and the parallel efficiency of the NEMO kernel respectively for the two benchmark configurations. *Figure 17* and *Figure 18* show their execution time trend, while speedup is reported in *Figure 19*.
We can note that the parallel efficiency quickly decreases when we use all the cores of the node in both cases. This is due to the increase of memory contention inside the node. We have another loss of performance running the MFS configuration on 64 cores, when the communication/computation ratio increases up to ~39%. The analysis of the results could suggest future code modifications that can be easily implemented and tested on the kernel and then extended to the whole code.

Table 6: *Performance data for MFS configuration (871 x 253 x 72 grid points).*

#MPI tasks	JPI	JPJ	JPK	Exec time	Parallel Efficiency
1	869	251	72	5.60E-01	100.00%
2	435	251	72	2.64E-01	106.06%
4	218	251	72	1.46E-01	95.89%
8	218	126	72	9.46E-02	74.00%
16	109	126	72	9.77E-02	35.82%
32	109	63	72	5.00E-02	35.00%
64	55	63	72	3.62E-02	24.17%
128	55	32	72	2.75E-02	15.91%
256	28	32	72	4.86E-02	4.50%
512	28	16	72	4.75E-02	2.30%
1,024	14	16	72	2.72E-02	2.01%

Table 7: *Performance data for Big configuration (5760 x 1440 x 32 grid points).*

#MPI tasks	JPI	JPJ	JPK	Exec time	Parallel Efficiency
1	5758	1438	32	1.11E+01	100.00%
2	2879	1438	32	4.79E+00	115.87%
4	1440	1438	32	2.56E+00	108.40%
8	720	1438	32	1.72E+00	80.67%
16	720	719	32	1.79E+00	38.76%
32	360	719	32	8.98E-01	38.63%
64	360	360	32	4.58E-01	37.87%
128	180	360	32	2.28E-01	38.03%
256	180	180	32	1.13E-01	38.37%
512	90	180	32	7.03E-02	30.84%
1,024	90	90	32	4.35E-02	24.92%

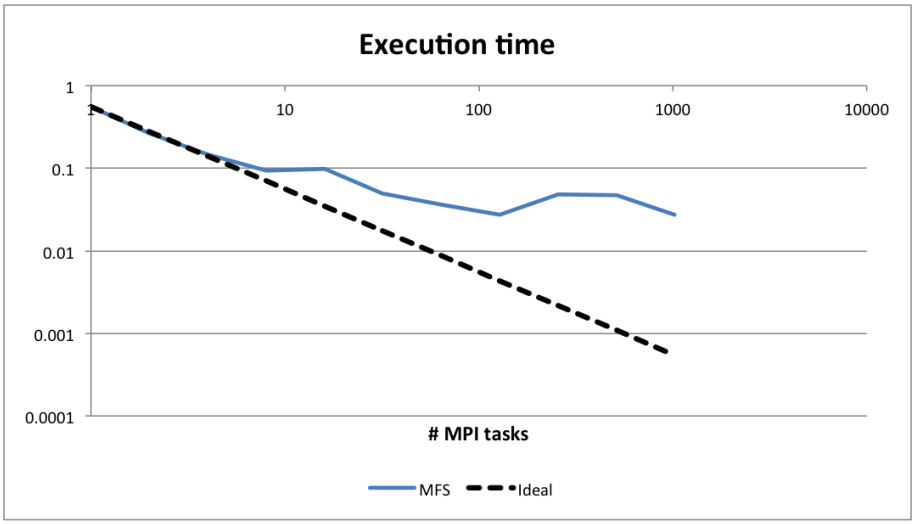

Figure 17: *MFS configuration execution time.*

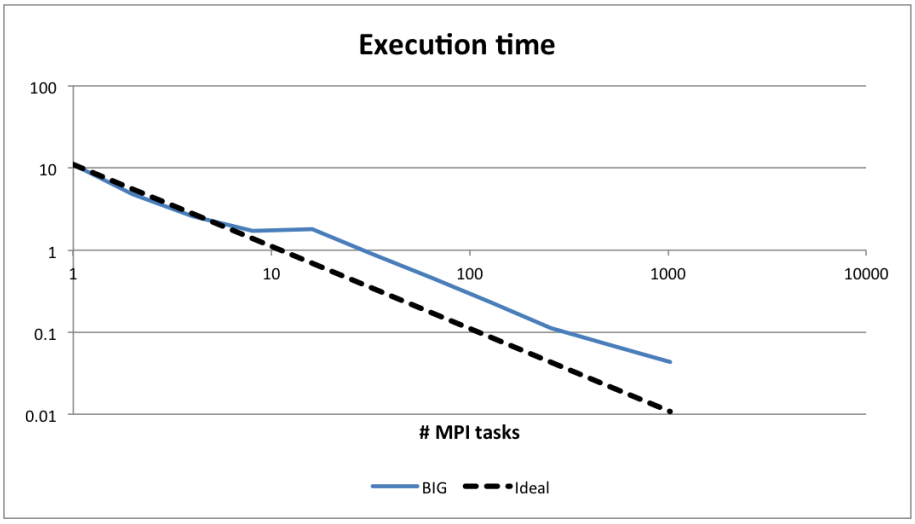

Figure 18: *Big configuration execution time.*

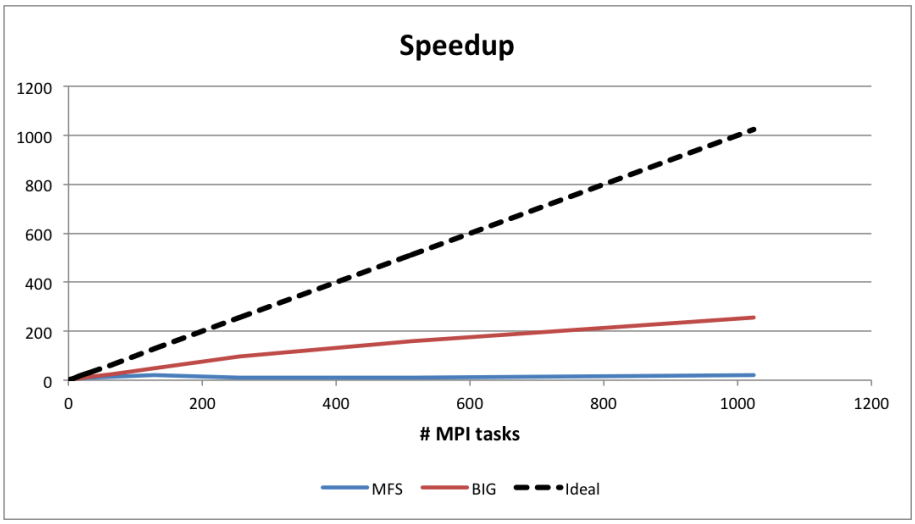

Figure 19: *MFS and Big configurations parallel speedup.*

5.2 ICON Communication Kernel

The purpose of the ICON communication kernel is to simplify further development and analysis of ICON communication. It focuses on the following features of the full model:

- Functionality to read ICON grid files
- Grid decomposition controlled by namelist variables
- Interoperability with other ICON kernels with communication requirements
- Features all relevant communication of the full model (relevance is determined by performance impact and required functionality by other ICON kernels)

The special role of this kernel (beyond performance, verification analysis and further development of ICON communication) consists in the desired ability to use it as software infrastructure for other kernels. Indeed, some parts of the model cannot be reduced to a single small stand-alone kernel because they require complex model functionality, e.g. iterative solver requires halo exchange. But including the halo exchange support would strongly inflate the solver kernel and weaken the focus. Therefore, the communication kernel has been implemented in a way that allows other ICON kernels to use it as software infrastructure, so that the functionality is available but the complexity is encapsulated.

Redmine project: https://redmine.dkrz.de/projects/icon-communication-kernel

Changes with respect to interim release (MS10.3)

None. This is a newly released kernel.

Instructions on download, execution and analysis

The source code of the ICON communication kernel is managed with a git repository. In order to get access to this repository or to download a snapshot of the current state, please contact behrens@dkrz.de.

Compilation of the kernel is explained in the README file in the top directory. An example ICON grid file and an example runscript are given in the run directory. The runscript is adapted to the SLURM batch system on Mistral. Adaption to other execution environments requires site-specific knowledge about executing MPI programs. The kernel can execute several tests. The selection is controlled by namelist parameters. Details are also described in the top level README file.

Depending on the given values for the namelist variable `testbed_set`, selected tests are executed one after the other within the same run. Each test produces a timer report which, depending on the set of active timers, reduces the timings of all MPI tasks to min, average, max and sum. Additionally the derived value `lbe` (load balance efficiency) is given as `t_avg/t_max` and estimates the load balance problem (100% equals perfect load balance in a simple resource utilization model).

Performance reference

The following examples show time measurements for varying node counts, using 36 processes per node.

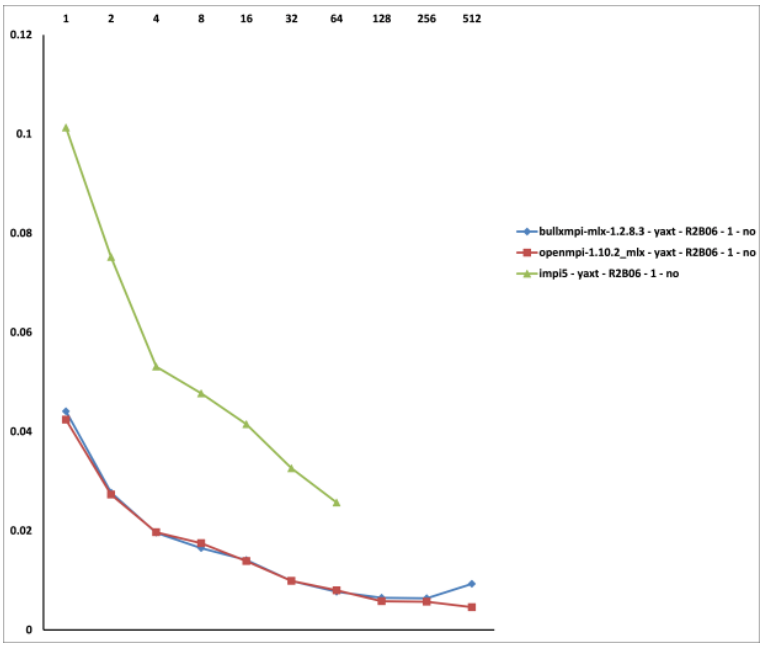

Figure 20: *Comparing MPI implementations. Using YAXT for the halo exchange of the R2B06 grid the openmpi based MPI implementations seem to be faster than the Intel version.*

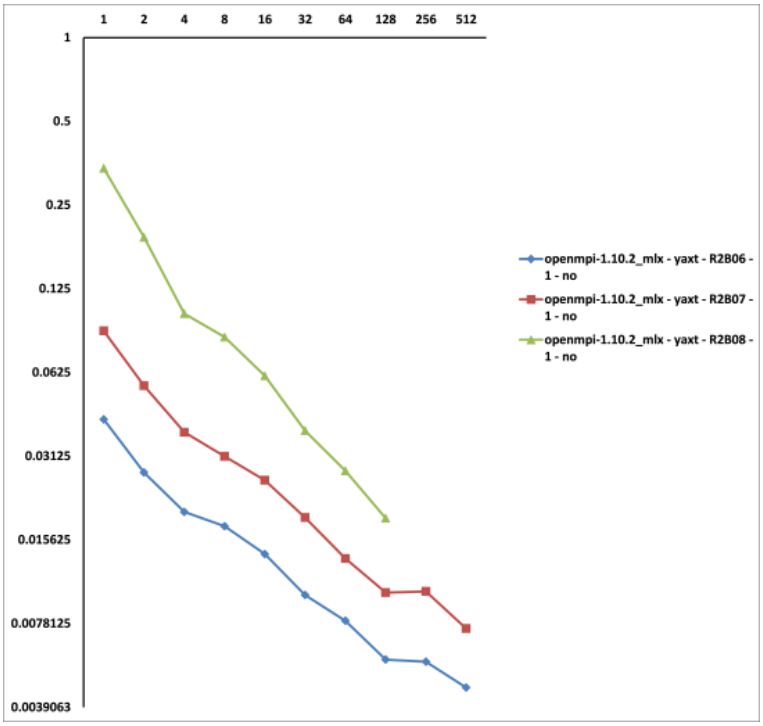

Figure 21: *Comparing different grid resolutions. YAXT scales well for varying grid resolutions.*

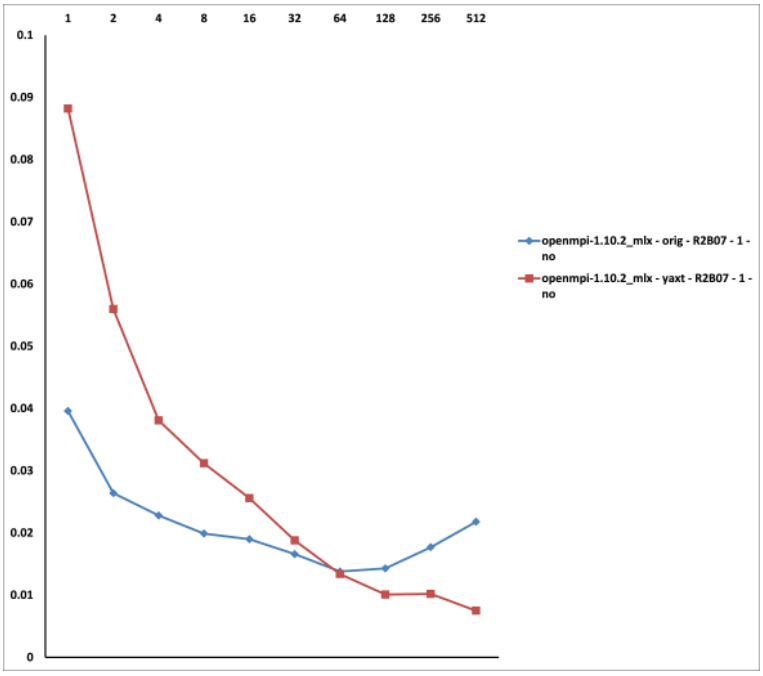

Figure 22: *Comparing YAXT implementation with original one. When comparing YAXT and the original implementation for exchanging the halos of a single 3D field, YAXT seems to perform better for large numbers of computing nodes, while the original implementation performs better for lower numbers.*

6. Outlook

Earth system models are evolving research codes that are undergoing steady changes. To stay relevant it is necessary to keep pace with the model developments and update the suite with newer versions of the benchmark codes and test cases. Since the benchmark releases are linked to the development and release cycles of the ESMs, it was unfortunately not possible to release benchmarks reflecting the final CMIP6 state of ESMs before the end of the IS-ENES2 project. However, several partners made commitments to continue the work on benchmark development beyond the IS-ENES2 project and make up-to-date benchmarks available.

The development of ESM benchmarks satisfying common benchmark requirements (i.e. complete, portable, well-documented, and verifiable) is a difficult task since evaluation of the correctness of execution is hardly possible on the basis of a short simulated time period. Therefore, currently only a limited number of prepared benchmarks provide some correctness checks. An objective, automatable methodology is needed for the verification of a correct benchmark execution to exclude errors due to oversights in porting, compiler bugs, too aggressive compiler optimisation etc. The ensemble-based consistency test proposed in [10] is a promising approach to implement such benchmark verification.

References

[1] Fogli, P.G. and D. Iovino: *CMCC–CESM–NEMO: toward the new CMCC Earth System Model*, Research Papers Issue RP0248, December 2014

[2] Information on NEMO: http://www.nemo-ocean.eu/

[3] CESM Software Engineering Group (CSEG): CESM User's Guide (CESM1.2 Release Series User's Guide). http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/book1.html

[4] Information on EC-EARTH Earth system model: http://www.ec-earth.org/

[5] Information on IPSL climate models: http://icmc.ipsl.fr/index.php/icmc-models

[6] Maisonnave, E. and A. Caubel: *LUCIA, load balancing tool for OASIS coupled systems*, Technical Report, TR/CMGC/14/63, URA CERFACS/CNRS No1875, France, 2014. http://pantar.cerfacs.fr/globc/publication/technicalreport/2014/lucia_documentation.pdf

[7] Information on MPI-ESM1 model: https://www.mpimet.mpg.de/en/science/models/mpi-esm/

[8] Information on ICON model: https://www.mpimet.mpg.de/en/science/models/icon/

[9] Madec, G. and the NEMO team: *NEMO ocean engine*, Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27 ISSN No 1288-1619, 2016. http://www.nemo-ocean.eu/About-NEMO/Reference-manuals/NEMO_book_3.6_STABLE/

[10] Baker, A.H., D. M. Hammerling, M. N. Levy, H. Xu, J. M. Dennis, B. E. Eaton, J. Edwards, C. Hannay, S. A. Mickelson, R. B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein, and D. Williamson: *A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0)*, Geosci. Model Dev., 8 2829-2840, 2015.

[11] Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., Caubel, A., Durachta, J., Foujols, M.-A., Lister, G., Mocavero, S., Underwood, S., and Wright, G.: *CPMIP: measurements of real computational performance of Earth system*, Geosci. Model Dev., 10, 19-34, 2017, http://www.geosci-model-dev.net/10/19/2017/, doi:10.5194/gmd-10-19-2017

Appendix A: HPC resources used to execute benchmarks from the ENES Benchmark Suite.

Table A.1: *System configuration of Athena at CMCC*

HPC system	Athena https://sccmon.cmcc.it
Organization	Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici (CMCC) https://www.cmcc.it
Vendor	IBM
Operational since	2013
Description	IBM System X iDataPlex DX360M4
CPU	Intel Xeon E5-2670 8 cores(Sandy Bridge)
Operating system	Linux CentOS 6 x86_64
Number of nodes	482
Cores per node	16
Number of cores	7712
CPU frequency	2.6 GHz
Memory per node	64 GB
Memory	30,1 TB
Peak performance	160 TFlop/s
Highest rank in TOP500 list	316 (November 2013)
Interconnect	InfiniBand 4x FDR
Batch system	LSF v. 8.0

Table A.2: *System configuration of Beskow at PDC*

HPC system	Beskow https://www.pdc.kth.se/resources/computers/beskow
Organization	PDC Center for High Performance Computing https://www.pdc.kth.se/
Vendor	Cray
Operational since	2014
Description	Cray XC40
CPU	Intel Xeon E5-2698 v3 16 cores (Haswell)
Operating system	Linux
Number of nodes	1676
Cores per node	32
Number of cores	53632
CPU frequency	2.3 GHz
Memory per node	64 GB
Memory	104.7 TB
Peak performance	1973 TFlop/s
Highest rank in TOP500 list	32 (November 2014)
Interconnect	Cray Aries (Dragonfly topology)
Batch system	SLURM

Table A.3: *System configuration of Curie at TGCC*

HPC system	Curie http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm
Organization	Très Grand Centre de calcul du CEA (TGCC) http://www-hpc.cea.fr/en/complexe/tgcc.htm
Vendor	Bull, Atos Group
Operational since	2012 (thin nodes), 2010-2016 (fat nodes), 2011-2016 (hybrid nodes)
Description	Cluster consisting of Bullx B510 (thin nodes), Bullx S6010 (fat nodes) and Bullx B505 (hybrid nodes)
CPU	Thin nodes: Intel Xeon E5-2680 8 cores (Sandy Bridge)
Operating system	Linux
Number of nodes	Thin nodes: 5040
Cores per node	Thin nodes: 16
Number of cores	Thin nodes: 80640
CPU frequency	Thin nodes: 2.7 GHz
Memory per node	Thin nodes: 64 GB
Memory	Thin nodes: 308 TB
Peak performance	Thin nodes: 1667 TFlop/s
Highest rank in TOP500 list	Thin nodes: 9 (June 2012)
Interconnect	InfiniBand QDR Full Fat Tree network
Batch system	SLURM

Table A.4: *System configuration of Ada at IDRIS*

HPC system	Ada http://www.idris.fr/eng/ada/
Organization	Institute for Development and Resources in Intensive Scientific Computing (IDRIS) http://www.idris.fr/eng/
Vendor	IBM
Operational since	2012
Description	IBM xSeries x3750 Cluster
CPU	Intel Xeon E5-2680 8 cores (Sandy Bridge)
Operating system	Linux
Number of nodes	332
Cores per node	32
Number of cores	10624
CPU frequency	2.7 GHz
Memory per node	128 GB, 256 GB
Memory	46 TB
Peak performance	233 TFlop/s
Highest rank in TOP500 list	123 (November 2012)
Interconnect	InfiniBand FDR10 Mellanox network (2 links per node)
Batch system	LoadLeveler

Table A.5: *System configuration of Mistral at DKRZ*

HPC system	Mistral https://www.dkrz.de/Klimarechner-en/hpc
Organization	German Climate Computing Center (DKRZ) https://www.dkrz.de
Vendor	Bull, Atos Group
Operational since	Phase 1: June 2015 Phase 2: June 2016
Description	Bullx DLC 720 blade cluster with Intel Xeon E5 processors
CPU	Phase 1: Intel Xeon E5-2680V3 12cores (Haswell) Phase 2: Intel Xeon E5-2695V4 18cores (Broadwell)
Operating system	Linux (RedHat)
Number of nodes	Phase 1: 1550 nodes Phase2: 1750 nodes
Cores per node	Phase 1: 24 Phase 2: 36
Number of cores	100200
CPU frequency	Phase 1: 2.5 GHz ; Phase 2: 2.1 GHz
Memory per node	64 GB, 128 GB, 256 GB
Memory	249 TB
Peak performance	3.6 PFlop/s
Highest rank in TOP500 list	33 (June 2016)
Interconnect	FDR InfiniBand (fat tree topology with a blocking factor 1:2:2)
Batch system	SLURM v 16.0.5