EUROPEAN COMMISSION

Community research

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

IS-ENES2 DELIVERABLE (D -N°: 9.4)

CDI-pio & XIOS I/O servers compatibility with HR climate models

Original title: **Benchmark of the two I/O servers**

{File name: IS-ENES2_D9.4.pdf}

Authors: **Eric Maisonnave, Irina Fast, Thomas Jahns, Joachim Biercamp, Stéphane Sénési, Yann Meurdesoif**

Reviewer: **Uwe Fladrich**

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

Release date for review: ***09/03/2017***

Final date of issue: ***16/03/2017***

Revision table			
Version	**Date**	**Name**	**Comments**
1	01/03/17	Eric Maisonnave	First draft
2	07/03/17	Eric Maisonnave	Reformatting
3	09/03/17	Irina Fast, E.M.	Corrections, reformatting
4	16/03/17	Eric Maisonnave	Final version

Abstract

I/O performance is likely a leading constraint to efficient operation of High Resolution Earth System Model (HR ESM) simulations. The I/O libraries CDI-pio (jointly developed by MPG and DKRZ) and XIOS (developed at IPSL) were implemented and tested with several components of the earth system models used in the present IS-ENES2 work-package 9. CDI-pio was evaluated with the atmospheric model ECHAM6 and XIOS with the atmospheric model ARPEGE-Climat and the ocean model NEMO. Both CDI-pio and XIOS2 I/O libraries prove their capacity to efficiently manage high resolution model output.

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

- ***Objectives***

I/O performance is likely a leading constraint to efficient operation of High Resolution Earth System Model (HR ESM) simulations. Several IS-ENES partners evaluated the work associated with the implementation of an I/O library, which makes possible the data output on dedicated resources (I/O servers), into their respective model and assess performance benefits in stand alone components, coupled systems and Multi-Model High Resolution setting. The I/O libraries CDI-pio (jointly developed by MPG and DKRZ) and XIOS (developed at IPSL) were implemented and tested with several components of the earth system models used in the present IS-ENES2 work-package 9. CDI-pio was evaluated with the atmospheric model ECHAM6 and XIOS with the atmospheric model ARPEGE-Climat and the ocean model NEMO.

- ***Results***

Both CDI-pio and XIOS2 I/O libraries prove their capacity to efficiently manage high resolution model output. With ECHAM-XR, at high data intensity, CDI-pio keeps the data output cost at reasonable level (~6%). With NEMO-ORCA025, within the IS-ENES2 demonstrator ESM (HiResMIP experimental setup), the data output cost is about 1%. More demanding data intensity conditions (up to 1.5 GB/CH) obtained with the ideal case GYRE-144, leads to satisfactory output cost, between 1.5% (daily output) to 20% (hourly output).

- ***Perspectives***

Switching from serial to parallel output in ECHAM6 via CDI-pio library significantly reduced computing resources required to perform data-intensive high resolution simulations with MPI-ESM at scale. Efforts to implement CDI-pio in MPIOM (the ocean component of MPI-ESM) and ICON, an icosahedral non-hydrostatic earth system model, are ongoing now. XIOS, used by a large variety of climate models, will be deployed on new models (Met Office).

1. IO server compatibility with HR climate models

I/O performance is likely a leading constraint to efficient operation of High Resolution Earth System Model (HR ESM) simulations. Several IS-ENES partners evaluated the work associated with the implementation of I/O server libraries into their respective model and assessed performance benefits in standalone components, coupled systems and Multi-Model Multi-Member High Resolution (M4-HR) setting. In the present document, recent developments of the I/O libraries CDI-pio and XIOS (from DKRZ/MPG and IPSL) are summarised. We also describe their implementations in several components of the climate models used in the present IS-ENES2 work-package 9, from which the ECHAM and ARPEGE-Climat atmosphere model, and the NEMO ocean model.

Table 1 summarizes the implementations that will be described in the document.

***Table 1**: I/O libraries implemented in IS-ENES2-related HR ESM*

Component	ESM	Library	Implementation
ARPEGEv6	CNRM-CM6	XIOS2	CNRM
ECHAMv6.3	MPI-ESM1	CDI-pio	MPG and DKRZ
NEMOv3.6	ARPEGEv5-NEMO	XIOS1	IPSL
	EC-Earth3	XIOS2	EC-Earth consortium

We also present the computing performance of XIOS implemented in one component NEMO of our ESMs. To be able to evaluate these performances, two main metrics were chosen: *Data Output Cost* and *Data Intensity*, as defined in Balaji et al 2017. The Data Output Cost gives a measure (%) of how much of additional resources (runtime*cores) is necessary when model output is activated. The Data Intensity gives an idea of how much data is produced per core hour (GB/CH). This number strongly depends on the experimental design: the more data is written to disk, the higher is the Data Intensity. But this metric is limited by the Data Output Cost (core hours can increase when data are produced). An I/O library proves its efficiency in a HR ESM context when Data Output Cost is low and Data Intensity is high (i.e. a lot of data is produced with a small extra cost). The three supercomputers used are described in Table 2.

***Table 2**: Supercomputers characteristics*

Machine	Chip	Cores	Clock speed (GHz)	Interconnect	URL
curie (thin nodes)	Sandy Bridge	80 640	2.7	InfiniBand QDR	http://goo.gl/RR5
mare nostrum III	Sandy Bridge	48 896	2.6	Infiniband FDR	https://www.bsc.es/innovation-and-services/supercomputers-and-facilities/marenostrum
mistral (phase 1)	Haswell	37200	2.5	Infiniband FDR	https://www.dkrz.de/Nutzerportal-en/doku/mistral/configuration

2. CDI-pio

2.1 Overview

The **CDI** (**C**limate **D**ata **I**nterface) library provides a machine and format independent interface for reading and writing of data stored in common scientific data formats used in numerical weather prediction and climate modelling. CDI is broadly employed in models developed at Max-Planck-Institute for Meteorology (MPG) to allow for a flexible configuration of the model output. The original version of the library supported multi-thread, single-process semantics only. Over time, the lack of MPI-IO support in CDI had become one of the major performance bottlenecks on massively parallel systems with distributed memory. **CDI-pio** is a parallel extension of the serial CDI library. It provides an I/O server infrastructure for writing data to disk via dedicated I/O processes (asynchronous output). This technique is combined with the parallel I/O (i.e. collective write of data by multiple processes to shared, on-disk files). CDI-pio can handle GRIB1, GRIB2, all variants of netCDF (classic format, 64-bit offset format, netCDF-4 format, and netCDF-4 classic model format). Support of various data formats is a unique feature of CDI-pio compared to other high-level parallel I/O libraries worldwide. Efficient data compression capabilities supported by CDI can considerably reduce the data output volume.

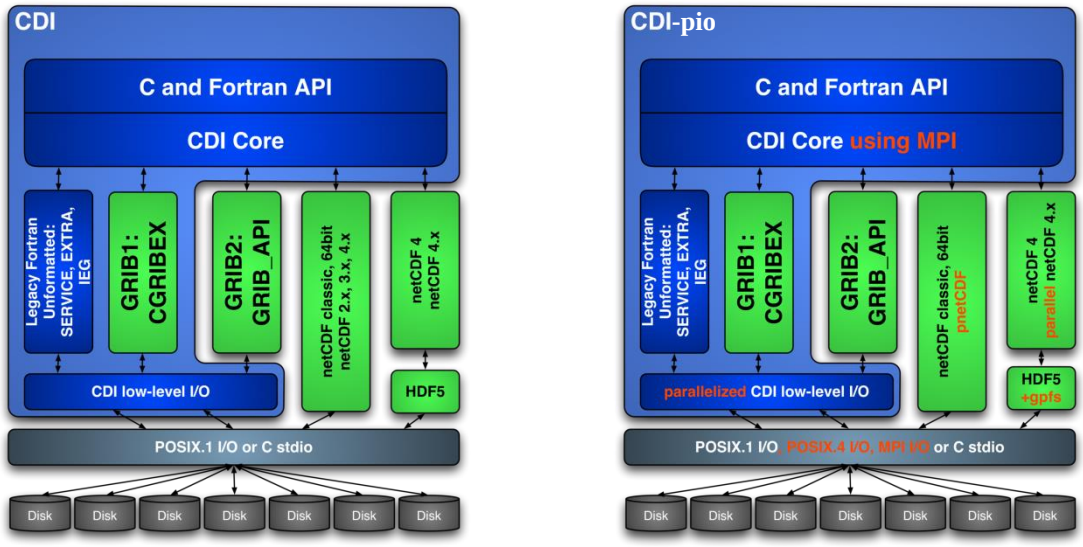


Figure 1: *Comparison of serial (left, CDI) and parallel (right, CDI-pio) libraries. The used backend data format libraries are displayed by green boxes. Extensions relevant to parallel output are highlighted in red.*

Parallel I/O of netCDF files uses PnetCDF (parallel netCDF) library for classic XDR-based formats and parallel netCDF4/HDF5 libraries for more recent HDF5-based format. Parallel I/O for GRIB and binary data formats is realized with MPI-IO routines[1]. Serial low level writing modes using POSIX I/O or C stdio are also implemented for asynchronous output.

[1] MPI_FILE_IWRITE_SHARED, MPI_FILE_WRITE_AT, MPI_FILE_WRITE_ODERED, or MPI_FILE_WRITE_AT_ALL depending on selected I/O mode

Depending on the hardware architecture, the file system, record size and the MPI implementation some can provide a better performance than others.

I/O servers collect, transpose, encode, compress, buffer, align, and write data to on-disk files. Output data is collected via RDMA to avoid congestion of the communication system and disturbance of the processes reserved for the model computation tasks. The transposition of data from any model decomposition to output decomposition matching the file layout is accomplished with the generic **YAXT** (**Y**et **A**nother e**X**change **T**ool) data redistribution library developed at DKRZ.

2.2 Recent developments and future work

Since the initial release of CDI-pio in 2012, further work was focused on the following functional aspects:

- Implementation of additional parallel output modes, accounting for different patterns
- Support for unstructured grids
- Improved support for netCDF, especially parallel output capabilities
- GRIB2 support
- Distributed meta-data API extension

Some effort was expended to identify performance drawbacks, enhance the scalability and achieve better I/O bandwidth. After analysis of current performance was performed, the following optimisations were implemented:

- Optimised algorithms for computation of YAXT communication patterns (redists)
- Caching of YAXT re-usable redists
- Improvements in memory management
- Eliminate redundant overwrites of client buffers
- Algorithmic improvements for the communication of client and server processes

The following topics are currently in progress or designated for future implementation:

- Introduction of OpenMP parallelization
- Server-less mode for models that benefit more from running on all tasks vs. delegating I/O to extra tasks
- Support for parallel I/O with binary data formats (SERVICE, EXTRA, IEG) developed at MPG
- Automatic determination of optimal data chunks in all dimensions

Most of the above work was done in the context of the German HD(CP)2 project or on internal DKRZ funding and is included here for the sake of completeness.

2.3 How to use the CDI-pio library with an Earth System Model (ESM)

To use all available features of the CDI-pio the following additional libraries are required:

- MPI
- NetCDF *MPI-enabled*
- HDF5 *MPI-enabled*
- PnetCDF
- YAXT
- ScalES-PPM

References for download of source distributions of all needed libraries are provided in the Download section.

CDI and CDI-pio are written in C and provide C- and Fortran-APIs for use in ESMs. By design, the CDI-pio API retains close compatibility with the established CDI API. Therefore, little adaptation effort is necessary for applications already using CDI for data output. CDI manuals describe how to build and install CDI library from the sources on UNIX-like systems, document CDI C- and Fortran-APIs and provide complete examples for writing and reading a dataset using CDI. Necessary extensions to implement asynchronous parallel output via CDI-pio are described in the CDI-pio manual. It is also advisable to take a look at examples of CDI-pio usage included in the CDI package since the information in the manual is not always up-to-date. The C and Fortran examples demonstrate a typical sequence of YAXT and CDI-pio calls with a good measure of relevant details.

From the user perspective, the necessary implementation tasks are:

1. Initialize YAXT and CDI libraries
2. Generate CDI-pio configuration object and select write mode and distribution of I/O servers over nodes
3. Split the global communicator into compute tasks and I/O tasks, update model communicator
4. Generate description of domain decomposition using YAXT library
5. Create and define metadata objects (horizontal grid, vertical axes, time axes)
6. Create a variable list, define variables, their names and declare their associated horizontal grids, vertical axes and attributes.
7. Assign the time axis to the variable list
8. Open a file for a data set
9. Assign the defined variable list to the data set (file)
10. Define date and time and write data to the file in the model time loop
11. Close the file
12. Clean up (destroy generated objects)
13. Finalize YAXT and CDI libraries at the end of model integration

CDI-pio provides library calls for each of the above steps, especially rich meta-data to produce self-describing data formats usable by many data processing software packages.

Currently, CDI-pio provides eight different low-level write modes:

• **PIO_NONE**	0	One process collects, encodes, compresses, buffers and writes using C `fwrite`
• **PIO_MPI**	1	All processes collect, gather, encode, compress, buffer and write using `MPI_FILE_IWRITE_SHARED`
• **PIO_ASYNCH**	2	One process writes the files using low level POSIX `aio_write`, the others collect, gather, encode, compress and buffer
• **PIO_FPGUARD**	3	One process guards the file offsets, all others collect, gather, encode, compress and write using C `fwrite`
• **PIO_WRITER**	4	One process writes the files using C `fwrite`, the others collect, gather, encode, compress and buffer
• **PIO_MPI_FW_ORDERED**	5	All processes write relying on the MPI shared file pointer collectively
• **PIO_MPI_FW_AT_ALL**	6	All processes write collectively and unaligned, exchanging offset information only
• **PIO_MPI_FW_AT_REBLOCK**	7	All processes write individually but aligned to the underlying file system, exchanging amounts of data and parts of data as needed

The most appropriate choice of I/O mode, number and location of I/O processes to achieve the best performance on a specific machine needs to be determined by test runs. No additional configuration files are needed for CDI-pio. Nonetheless, the CDI API enables reconfiguration of output (e.g. data format, output fields, output frequency, I/O method, number and distribution of I/O servers) at run time.

2.4 Performance benchmarks

To measure the CDI-pio performance we defined a benchmark based on the global ECHAM-6.3 model which is the atmospheric part of the MPI-ESM1 model developed at MPG. The measurements were performed on the phase 1 of the HPC system 'Mistral' (https://www.dkrz.de/Nutzerportal-en/doku/mistral) installed at DKRZ in July 2015. It consists of 1550 Bullx B700 DLC nodes. Each node is equipped with two Intel Xeon E5-2680 v3 12-core Haswell processors with the nominal clock-rate of 2.5 GHz. For storage the Lustre parallel file system is used.

We consider two model configurations intended to be used in CMIP6-endorsed MIPs (DCPP and HighResMIP). The configuration ECHAM-HR (T127L95 grid) has a horizontal mesh with 192 x 384 grid cells. The resolution of the second configuration ECHAM-XR (T255L95) is doubled in longitudinal and latitudinal directions. The horizontal grid counts 384x768 grid cells with approx. 50 km grid point distance. The number of vertical levels (95) is the same for both configurations.

The simulated time period has been set to 1 month. By default, the output data are grouped in 10 data sets (corresponding to 10 files) and written in GRIB format. Table 1 summarizes further output specifics (number of 2D and 3D fields, output frequencies for different data sets etc.). The total data volume per simulated month is 22 GB for HR and 86 GB for XR setups. The measurements were performed in uncoupled mode to eliminate the effect of load imbalance between atmosphere and ocean model components in the coupled MPI-ESM model.

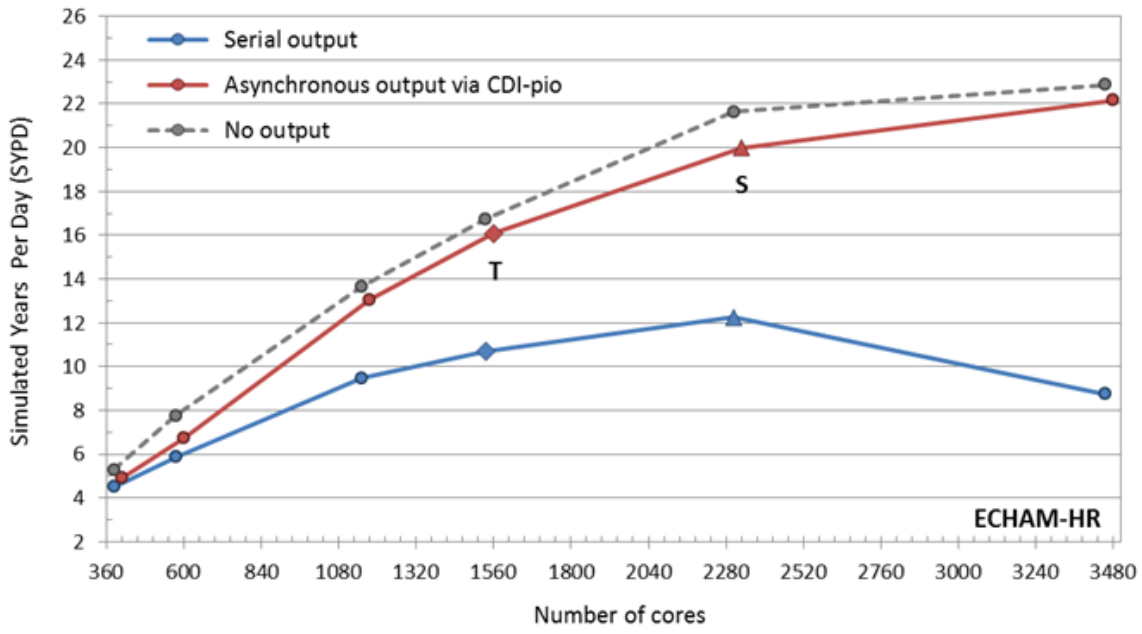
For benchmarking CDI release 1.7.0 and YAXT release 0.4.4 have been used. The ECHAM model and libraries source was compiled with the Intel Fortran/C compiler version 16.0.3. Bull X MPI 1.2.8.3 with Mellanox acceleration libraries MXM 3.3.3002 and FCA 2.5.2393 provided for communication. At run time, one MPI task per core (i.e. no hyperthreading) was configured.

We present here the out-of-the-box performance that can be achieved without MPI, network, and file system specific tuning efforts (e.g. by specifying of RDMA or ROMIO hints etc.). The measurements were made for a range of different domain decompositions to capture the scalability characteristics of CDI-pio. For each number of cores we run three output scenarios: serial output, parallel output via CDI-pio and no output at all. The serial approach corresponds to the gathering of global arrays on the root process and then writing data to a file. Since compute nodes are always allocated exclusively on Mistral cluster we dedicated a whole node (i.e. 24 cores) for I/O servers even if that was not necessary at low scale.

***Table 3**: Details on model output configuration for ECHAM6-HR and ECHAM6-XR.*

File tag	Number of 3D fields	Number of 2D fields	Output period	Number of time records per file	File size HR (GB)	File size XR (GB)
accw	-	8	6 hours	123	0.06	0.25
co2	-	11	6 hours	123	0.25	1.01
echam	10 (nlev=95)	125	6 hours	123	13.40	53.27
echamm	7 (nlev=95)	39	1 month	1	0.07	0.26
jsbach	15 (nlev=11)	12	6 hours	123	1.21	4.83
land	1 (nlev=5)	23	6 hours	123	0.21	0.85
ma	-	14	1 hour	743	1.43	5.71
surf	-	9	6 hours	123	0.15	0.61
veg	38 (nlev=11)	36	6 hours	123	2.24	8.97
yasso	36 (nlev=11)	2	6 hours	123	2.54	10.17
Total	**17 / 89 / 1**	**279**			**21.57**	**85.95**

Figure 2 shows the scaling behaviour of the ECHAM model with serial and parallel output. For reference, throughput rates for a configuration without any output are also shown. Obviously, employing CDI-pio for output significantly reduces the total run time and improves the scalability of the model. Following the CPMIP protocol, we focus on two computational points of interest: throughput (T) mode selected for efficient resource allocation and speed (S), or capability mode selected for maximal speed (Balaji et.al. 2017). For ECHAM-HR configuration the number of the simulated years per day (SYPD) increases from 10.7 to 16.1 years in throughput mode and from 12.2 to 20 years in speed mode. This corresponds to an increase of parallel efficiency from 43% to 66% in S-mode. Similar performance benefits arise for XR. Model throughput rate rises from 2.3 to 3.8 SYPD in T-mode and from 2.8 to 4.3 SYPD in S-mode. Thus, the parallel efficiency increases from 37% to 56% in S-mode. Furthermore, due to better overall scalability, both S- and T-points could be moved to considerably higher number of cores, increasing the achievable productivity.



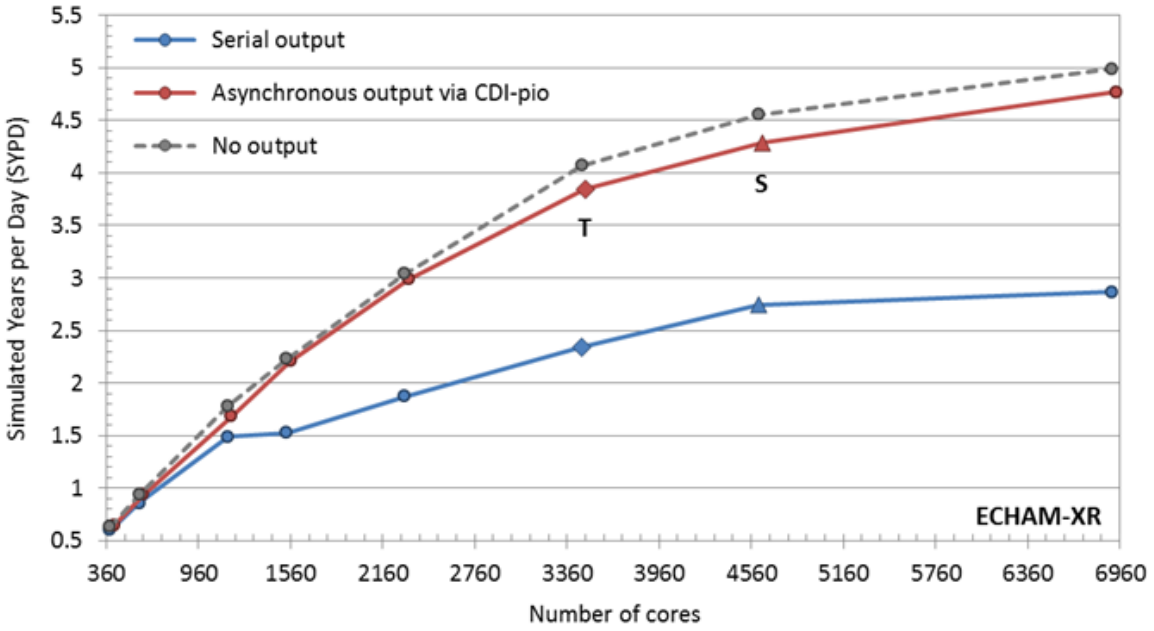


***Figure 2**: Simulated number of years per day (SYPD) for ECHAM-HR (T127L95, top) and ECHAM-XR (T255L95, bottom) configurations with serial output (blue), asynchronous parallel output via CDI-pio (red) and without output (dashed grey). The throughput (T) mode is marked by a diamond and the speed (S), or capability mode is marked by a triangle on the curves*

Table 4 shows further CPMIP metrics which demonstrate the advantages of using of CDI-pio for data output.

Table 4: *Computational and I/O cost for ECHAM-HR and ECHAM-XR with serial (left) and parallel (right) output. Measurements for throughput (T) and speed (S), or capability modes are listed. See Balaji et al, 2017 for a definition and discussion of metrics.*

Configuration	Resolution	SYPD		CHSY		Data Output Cost		Data Intensity	
ECHAM-HR T	7×10^6	10.7	16.1	3445	2329	36%	6%	0.075	0.111
ECHAM-HR S	7×10^6	12.2	20.0	4515	2793	43%	8%	0.057	0.093
ECHAM-XR T	28×10^6	2.3	3.8	35424	21727	43%	6%	0.029	0.047
ECHAM-XR S	28×10^6	2.8	4.3	40274	25939	40%	6%	0.026	0.040

Since the running cost of the benchmarks considered here is relatively high it was unfortunately not possible to provide the measurements with a confidence interval since this would require many repetitions of every test to have a statistically sound sample size. To assess the variability and reliability of reported values, we repeated some selected measurements on different days. The measured total run times show a spread by up to 20% due to dynamical changes of network and I/O load on the `mistral` system. Nevertheless, the overall relation between run times of triplets (serial output, parallel output, no output) presented here is well captured.

2.5 Download

In this section we provide URLs for complete documentation and download of freely available third-party libraries needed to resolve all CDI-pio dependencies.

- NetCDF data access libraries developed at the Unidata Program Center in Boulder: http://www.unidata.ucar.edu/software/netcdf/
- HDF5 library and tools developed by Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratory (SNL) and Los Alamos National Laboratory (LANL): https://support.hdfgroup.org/HDF5/
- PnetCDF parallel I/O library developed by Northwestern University and Argonne National Laboratory for netCDF file access http://trac.mcs.anl.gov/projects/parallel-netcdf/
- YAXT library developed at DKRZ for distributed memory programming: https://www.dkrz.de/redmine/projects/yaxt/
- ScalES-PPM library developed at DKRZ to provide parallelization and performance oriented modules particularly for ESMs: https://www.dkrz.de/redmine/projects/scales-ppm/

The MPI library is usually part of an HPC centers software stack. Various commercial and free implementations are available.

CDI-pio is part of the CDI distribution made available by MPG. It is freely distributed under the LGPL. Tar archives are available for download at https://code.zmaw.de/projects/cdi/ or from the git repository with

```
git clone https://git.mpimet.mpg.de/public/libcdi.git
```

3. XIOS

3.1 Overview

XIOS, standing for XML-IO-Server, is a library dedicated to I/O management in climate codes. XIOS manages output of diagnostics and other data produced by climate component codes into files and offers temporal (average, minimum, maximum, etc.) and spatial post-processing operations on this data. XIOS aims at simplifying the I/O management in the codes by minimizing the number of subroutine to be called in the code and reducing the number of arguments of these subroutines. The output definition is outsourced in an XML file and this allows easily changing the output configuration without recompiling. XIOS is fully parallel and targets climate models running on large number of cores. Its efficiency is based on the server concept into which one or more additional processes are exclusively dedicated to the asynchronous writing of the data into files.

3.2 Relevant developments

The I/O library developed at Institut Pierre Simon Laplace (IPSL) was enhanced during the IS-ENES2 project to better fit community user needs. A new release (XIOS2) was provided in 2015 and used into our models, from which the CMIP6-endorsed coupled systems IPSL-CM6 and CNRM-CM6. XIOS2 development was focused on new internal design, enabling more easily the implementation of new functionalities. Although XIOS2 syntax is not strictly backward compatible with XIOS1, changes are minor, so that the management of the 2 versions in the same model for an overlapping time was easy. Functionalities provided by the new version of XIOS are:

- New management of memory buffer for client-server : buffer size is evaluated at context definition closing and has not to be specified any more, except for memory or computing performance. Depending on the case, huge memory size can be saved (factor 2-5 or more)
- Enhanced calendar functionality : User defined calendar can be specified, time operation (duration and date manipulation : ex : date +duration-> new date, or duration-duration-> new duration) are available from FORTRAN API
- Management of netCDF4-HDF5 format or netCDF4 classical format using PnetCDF in parallel
- Management of time-series : each variable can be output automatically in a single file with a splitting defined independently of the chunk of simulation, i.e. a written file can be reopened and appended
- Field of any dimension are managed, the constraint of 2D or 3D was released, i.e. scalar, 1D, as well as 4D, and up, can be output with time records
- Data reading in the model[2] is now possible (file reading and not only file writing functionality). Data are read and sent asynchronously by the server. In case of periodic read (for example forcing file), the server will do pre-fetching, i.e. data will be sent and stored in the client in advance to be available when the model will request the data, avoiding reading latency
- Grid transformation is introduced, i.e. to transform field (provided by the model or read in input file) into a new field, changing dimension, parallel distribution and data. A large variety of transformation will be implemented in the future. Interpolation of vertical axis and horizontal domain, axis inversion are available. Parallel global or partial reduction like global mean, zonal mean, etc. are in development. All implemented transformation filters are (or will be) fully parallel and scalable. Many more transformation filters will follow, like gradient computation, vorticity, neighboring discovery, halo transferring management, etc.
- Internally, XIOS has been rewritten in term of parallel workflow and data flow. Data provided by model or read in input file are expressed in term of flow assigned to a

[2] using the subroutine `xios_recv_field` with argument “field ID”, and "field array".

timestamp. A flow can be connected to filters, creating a new modified flow in output. There are time filter (which cumulate and manipulate data for an amount of time), arithmetic filters which perform and combine any point to point arithmetic operations between several incoming flows, and grid transformations which change the shape of the flow. All filters can be chained to perform complex operations. The idea is to provide a way to simply implement "in situ" parallel pre- and post-treatment chain on allocated computing nodes. The most costly part of post-treatment could be done by this way very quickly and all along the running simulation.

3.3 Interface implementation on models

XIOS is implemented in several components of the French CMIP6 endorsed ESMs. The implementation in NEMO, started before the beginning of the IS-ENES2 project, increases the number of ESM which includes the I/O library[3]. Three of the HR ESM described in this document benefit from the XIOS capabilities: CNRM-CM6, ARPEGE5-NEMO and EC-Earth3. The last two climate models are part of the HR ESM demonstrator presented in IS-ENES2 deliverable 9.6 (Maisonnave et al. 2017). Only the ocean part of these models includes XIOS. The implementation, prior to the project start, is not described in this document. As an up-to-date example of how XIOS can be implemented in a climate model component, we describe here the ARPEGE/SURFEX XIOS interface.

In CNRM-CM6, both ARPEGE v6.3 (atmosphere) and SURFEX v8.1 (surface) are interfaced to XIOS2.

To use all available features of XIOS the following additional libraries are required:

- MPI
- NetCDF *MPI-enabled*
- HDF5 *MPI-enabled*
- OASIS *if coupled configuration*

References for download of source distributions of all needed libraries are provided in the Download section.

Because the SURFEX "stand-alone" mode must be autonomous in using XIOS, it has been interfaced first. When SURFEX is included in ARPEGE, SURFEX interface to XIOS must cope with all possible geometries for these models; hence it must be able to inherit grid description from ARPEGE; in that case, the SURFEX interface must also cope with the `NPROMA` vector-slicing scheme of ARPEGE, which is not handled natively by XIOS (which assumes that fields for the whole of the MPI task are delivered). Given that ARPEGE is no more used in climate runs without SURFEX, and given the capabilities of XIOS/SURFEX interface, there was no need to develop an autonomous XIOS/ARPEGE interfacing.

[3] like HadGEM-GC, CESM-NEMO, ACCESS-S, COSMO-NEMO, etc.

Hence, ARPEGE uses :

- all XIOS setups done by SURFEX, including the definition for XIOS domain corresponding to the whole grid, and including the calendar and time step setups,
- routine `sfx_xios_declare_fields` for declaring fields,
- routine `sfx_xios_send_block` for delivering fields,
- when needed, routine `set_axis` for declaring a new axis to XIOS

Scanning ARPEGE geometries for declaring it to XIOS[4] has a lot in common with what is needed for declaring it to OASIS. Mask description for the various tiles is located in SURFEX[5].

From the user perspective, activating XIOS is done modifying a SURFEX namelist. In addition, an XIOS configuration file named `iodef.xml` need to be provided to define output fields characteristics. All SURFEX usual diagnostics that can be produced are automatically known by XIOS following a declaration done through the API. Diagnostics are then automatically declared to XIOS, using their SURFEX name as an identifier and also as the netCDF variable name, and with netCDF attributes long name and units taken from the FORTRAN code. They are also automatically enabled, and their missing values and sampling period are set. The kind of XIOS time operation is set to 'instant', which matches the habits for most SURFEX diagnostics. If an identifier in XIOS configuration files does not match the name of a declared diagnostic, XIOS will raise an error if it is requested in a file. Activating XIOS for ARPEGE is done by activating the XIOS output scheme in SURFEX.

3.4 Performances with HR models

We present here the computing performance of the libraries implemented in one component NEMO of our ESMs. Due to a lack of time, performances of ARPEGE in HR configuration cannot be presented here[6]. To test XIOS in an ever demanding configuration than NEMO ORCA025, 91 vertical levels, which is the ocean part of the two ESM included in the IS-ENES2 HR demonstrator (ARPEGE5-NEMO and EC-Earth3), the results of an ideal test case (NEMO only in GYRE configuration) are also provided Table 5 summarizes the characteristics of the components which I/O performance is evaluated in the document. Resolution unit is given in million grid points. NEMO, as a component of the demonstrator models, were tested on `mare nostrum`, and NEMO, in the GYRE ideal configuration, on `curie`.

[4] this common part stand in routine `aroini_surf`
[5] in routine `mse/internals/sfx_xios_setup_aro` or in surfex/`OFFLIN/sfx_xios_setup_ol` for the SURFEX stand-alone case
[6] The development of CNRM-CM6 HR configuration is scheduled following CMIP6 agenda

***Table 5**: Characteristics of components that include one of the two tested I/O library*

Component	ESM	Component resolution	Version
NEMO	ARPEGE5-NEMO	134 x 10^6	3.2
	EC-Earth6	113.4 x 10^6	3.6
NEMO	GYRE (ideal)	386 x 10^6	3.4

a) XIOS on NEMO (IS-ENES2 demonstrator)

For both ARPEGE5-NEMO and EC-Earth3 cases, the time period of the simulations performed to measure the XIOS performance is set to 1 month. The amount of output data is reduced to a dozen of monthly and daily 2D fields, in netCDF format. The measurements were performed in coupled mode to reproduce the realistic setup of a CMIP6 HiResMIP experiment, but NEMO is the single component which uses XIOS to output its diagnostics. Demonstrator simulations were performed in throughput mode. For further details about the IS-ENES2 demonstrator experimental setup, please refer to deliverable 9.6 (Maisonnave et al, 2017). We present in Table 6 the out-of-the-box performance that can be achieved without specific tuning effort. Please notice that computational and I/O metrics (except resolution) are related to the whole coupled system and not to the single component which uses the I/O server.

Table 6: *Computational and I/O cost for ARPEGE5-NEMO and EC-Earth6*

Configuration	Model SYPD	Model CHSY	Data Output Cost	Data Intensity
ARPEGE5-NEMO	3.5	2783	1.5%	0.0053
EC-Earth3	1.2	14521	1.1%	0.0045

b) XIOS on NEMO (GYRE idealized case)

NEMO resolution (ORCA025) presented on the previous paragraph is in the range of resolution of components used in High Resolution projects (i.e. HiResMIP). To estimate XIOS capacity to fulfill requirements of higher resolution configurations (close to global ORCA12 resolution), the same metrics are applied to test case simulations carried with the "rectangular basin" configuration (GYRE) of the ocean model. Three sets of simulation are performed with a varying output frequency (daily, 6-hourly and hourly frequency). The model is decomposed on 8,160 MPI sub-domains and the parallel I/O servers spread on 32 cores (128 for hourly output frequency). Results are presented in Table 7.

Table 7: *Computational and I/O cost for GYRE-144*

Output frequency	Data Output Cost	Data Intensity
day	1.5%	0.0764
6-hour	5%	0.3700
hour	20%	1.4700

3.5 Download

XIOS is made available by IPSL. It is freely distributed under the CeCILL2. Sources are available from the cvs repository with

```
svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk XIOS
```

4. Conclusion

4.1 CDI-pio perspectives

Switching from serial to parallel output in ECHAM6 significantly reduced computing resources required to perform data-intensive high resolution simulations with MPI-ESM at scale. Efforts to implement CDI-pio in MPIOM (the ocean component of MPI-ESM) and ICON atmosphere and ocean models are ongoing now. The prototype implementation of CDI-pio in ocean model MPIOM has passed initial tests, but needs further tuning to achieve the envisioned performance goals.

4.2 XIOS library developments

XIOS will continue its development in the next few years, funded by two European Center of Excellence EoCoE and ESIWACE and the French ANR project CONVERGENCE. Several enhancements are scheduled, from which:

- Relaxing some constraints specific to climate, to make XIOS more versatile and usable for other scientific communities for HPC simulations (EoCoE).
- Maintaining and improving the XIOS support and training (ESIWACE)
- Improving the XIOS workflow functionalities by adding new kind of filters (CONVERGENCE)
- Making XIOS fully multithreaded in order to improve performance when linked with OpenMP models, adapted to many core future architectures (ESIWACE)
- Adding new coupling functionalities, and so, merging workflow, I/O and coupling functionalities into a single more flexible and standardized tool (ESIWACE)
- Adding GRIB2 format for output

XIOS is now well recognized and used by a large variety of climate models, (i) at IPSL, in the full earth system model (NEMO ocean, LMDZ atmosphere, ORCHIDEE land surface, INCA chemistry) and the new icosaedral grid model DYNAMICO, (ii) at Météo France/CNRM, in the full climate model (NEMO ocean, ARPEGE atmosphere, SURFEX surface, GELATO sea-ice), (iii) in the CROCO consortium (costal & regional ocean ROMS & MARS3D), (iv) at Met Office, in the MONC atmosphere and (v) at LGGE, in the MAR regional atmosphere. The I/O library will be deployed at Met Office in the next generation atmosphere model LFRIC and in the Atmospheric Component of the Unified Model, including the management of ensemble diagnostics through XIOS.

Glossary

ACCESS-S:	BoM/CSIRO global climate model
ARPEGE:	Météo-France/CNRM atmosphere model (spectral dynamics)
CESM-NEMO:	CMCC global climate model
CH:	Core.Hours
CNRM-CM:	Météo-France/CNRM climate model
COSMO-NEMO:	DWD ocean-atmosphere regional model
CPMIP:	**C**omputational **P**erformance **M**odel **I**ntercomparison **P**roject, see Balaji et al, 2017
CROCO:	Coastal and Regional Ocean COmmunity, https://www.croco-ocean.org/
DCPP:	**D**ecadal **C**limate **P**rediction **P**roject): https://www.wcrp-climate.org/modelling-wgcm-mip-catalogue/modelling-wgcm-cmip6-endorsed-mips/index.php?t&view=article&id=1065
DKRZ:	Deutsches Klimarechenzentrum, Hamburg
DWD:	Deutscher Wetterdienst, Offenbach
EC-Earth:	ECMWF based climate model, climate modeling community https://www.ec-earth.org/
ECHAM:	MPG atmospheric general circulation model, http://www.mpimet.mpg.de/en/science/models/echam/
ESIWACE:	Centre of Excellence in Simulation of Weather and Climate in Europe (H2020 project)
GB:	Giga Byte
HadGEM-GC:	Met-Office/NCAS global climate model
HDF5:	Hierarchical Data Format, geophysics data format
HighResMIP:	**High-Res**olution **M**odel **I**ntercomparison **P**roject, https://www.wcrp-climate.org/modelling-wgcm-mip-catalogue/modelling-wgcm-cmip6-endorsed-mips/index.php?t&view=article&id=1068
ICON:	ICOsahedral Non-hydrostatic earth system model jointly developed by DWD and MPG
IPSL:	Institut Pierre Simon Laplace, Paris
LGGE:	Laboratoire de glaciologie et géophysique de l'environnement, Grenoble
MIP:	**M**odel **In**tercomparison **P**rojects
MPI:	Message Passing Interface, communication library
MPI-ESM (ECHAM):	http://www.mpimet.mpg.de/en/science/models/mpi-esm/
MPI-OM:	MPG ocean model
MPG:	Max Planck Institute, Hamburg
NEMO:	Community ocean model, www.nemo-ocean.eu
NetCDF and **PnetCDF**:	(Parallel) Network Common Data Form, geophysics data format
OASIS:	Community coupling library, http://verc.enes.org/oasis/
RDMA:	**R**emote **D**irect **M**emory **A**ccess: a network hardware ability to directly access data stored in memory on a remote node without involving the CPUs on that node
ScalES:	**Sca**lable **E**arth **S**ystem Models project, https://www.dkrz.de/Klimaforschung-en/dkrz-und-klimaforschung/infraproj/scales/scales-en
SURFEX:	Météo-France/CNRM surface model
SYPD:	Simulated Year Per Day
YAXT:	**Y**et **A**nother e**X**change **T**ool

References

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., 2017: CPMIP: Measurements of Real Computational Performance of Earth System Models in CMIP6 , Geosci. Model Dev., 46, 19-34, doi:10.5194/gmd-10-19-2017

Maisonnave, E., Manubens, D., Fladrich, U., Seland, Ø., 2017: Multi-model multi-member integrated simulation: Computing performances of a demonstrator and other single high-resolution climate models, Technical Report, CECI, UMR CERFACS/CNRS No5318, France