Computational aspects and performance evaluation of the IFS-XIOS integration

Xavier Yepes-Arbós, BSC
Mario C. Acosta, BSC
Gijs van den Oord, NLeSC
Glenn Carver, ECMWF

18th workshop on high performance computing in meteorology, ECMWF, UK

24/09/2018

Index

1. Introduction

eScience center

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Introduction

- Earth system models have benefited of the exponential growth of supercomputing power
- This allows to use more complex computational models to find more accurate solutions
- As a consequence, the generated amount of data has grown considerably
- However, since the I/O was not significant enough in the past, not much attention was paid to improve it

Introduction

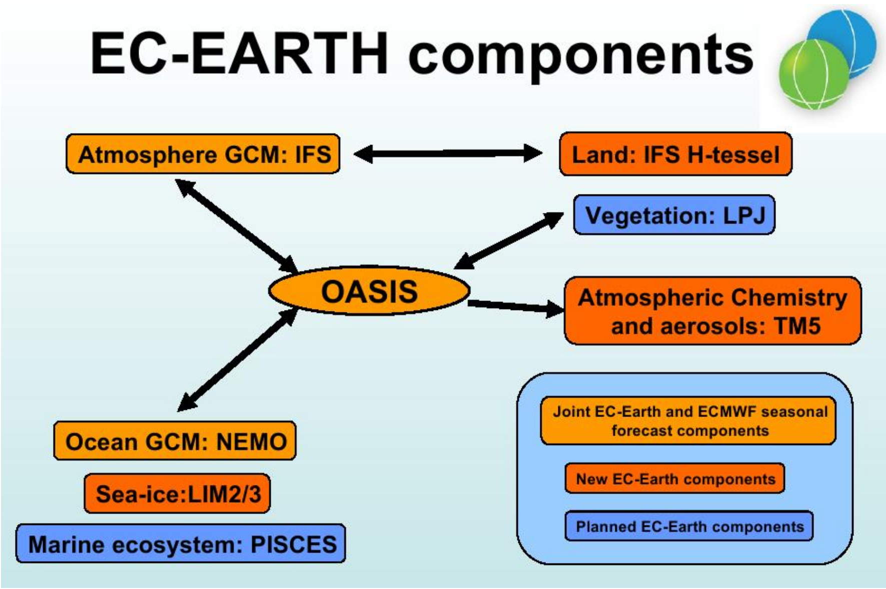
- Due to this reason, some Earth system models output data using inefficient sequential I/O schemes
- This type of scheme requires a serial process:
  - Gather all data in the master process of the model
  - Then, the master process sequentially writes all data
- This is not scalable for higher grid resolutions, and even less, for future exascale machines
- This is the case of one of the I/O schemes of IFS

Introduction

- IFS is a global forecasting system developed by ECMWF
- It has two different output schemes:
  - The Météo-France (MF) I/O server which is fast and efficient from a computational point of view. It is only used at ECMWF, such its operational forecasts
  - A sequential I/O scheme which is slow and inefficient from a computational point of view. It is used by non-ECMWF users, this is, in OpenIFS and in the IFS version of EC-Earth
- IFS is also used as a component of the EC-Earth model

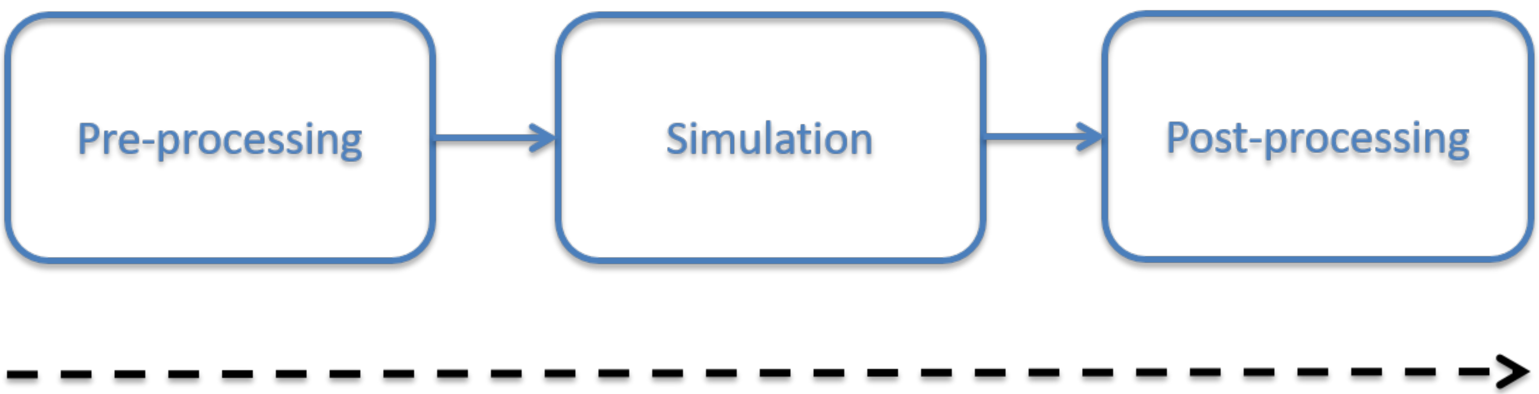
Introduction

- EC-Earth is a global coupled climate model, which integrates a number of component models in order to simulate the Earth system
- The two main components are IFS as the atmospheric model and NEMO as the ocean model



Introduction

- In addition, Earth system models such as EC-Earth, run experiments that have other tasks in their workflow
- Post-processing task can perform data format conversion, compression, diagnostics, etc.



Critical path = Pre-processing + Simulation + Post-processing

Introduction

- When IFS is used in EC-Earth for climate modeling, post-processing is needed to:
  - Convert GRIB files to netCDF files
  - Transform data to be CMIP-compliant (CMORization)
  - Compute diagnostics
- Post-processing turns into an expensive process

Motivation

- In particular, we are experiencing an I/O bottleneck in the IFS version of EC-Earth
- EC-Earth has been recently used to run experiments using the T511L91-ORCA025L75 configuration under the H2020 PRIMAVERA project
- Experiments require to output a lot of fields, causing a considerable slowdown in the EC-Earth execution time
- I/O in IFS represents about 30% of the total execution time

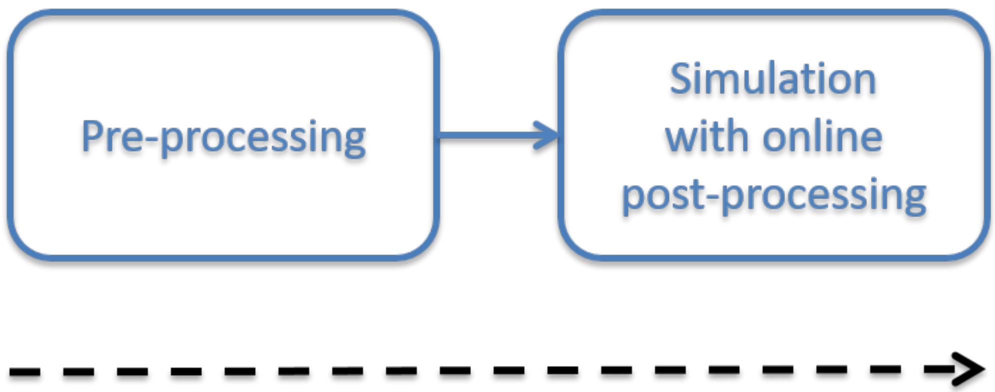
Motivation

- In order to address the I/O issue, we have to select a suitable tool that fulfills a series of needs:
  1. It must be a parallel, efficient and scalable I/O tool
  2. Data must be written using netCDF format (standard in climate modelling) and must follow the CMIP standard
  3. It must perform online post-processing along with the simulation, such as interpolations or data compression
- There is a tool designed to that end: XIOS
- XIOS is an I/O server

Motivation

The use of a tool such as XIOS has a twofold effect:

- Improve the computational performance and efficiency of a model, and thus, reduce the execution time
- Reduce the critical path of its workflow by avoiding the post-processing task



Critical path = Pre-processing + Simulation

European collaboration

- Netherlands eScience Center (NLeSC)/Koninklijk Nederlands Meteorologisch Instituut (KNMI)
- European Centre for Medium-Range Weather Forecasts (ECMWF)

2. Components description

IFS

- The Integrated Forecast System (IFS) is a global data assimilation and forecasting system developed by ECMWF
- It writes using the GRIB format (standard in weather forecast)
- It can use two different output schemes:
  - The MF I/O server
  - A sequential I/O scheme

XIOS

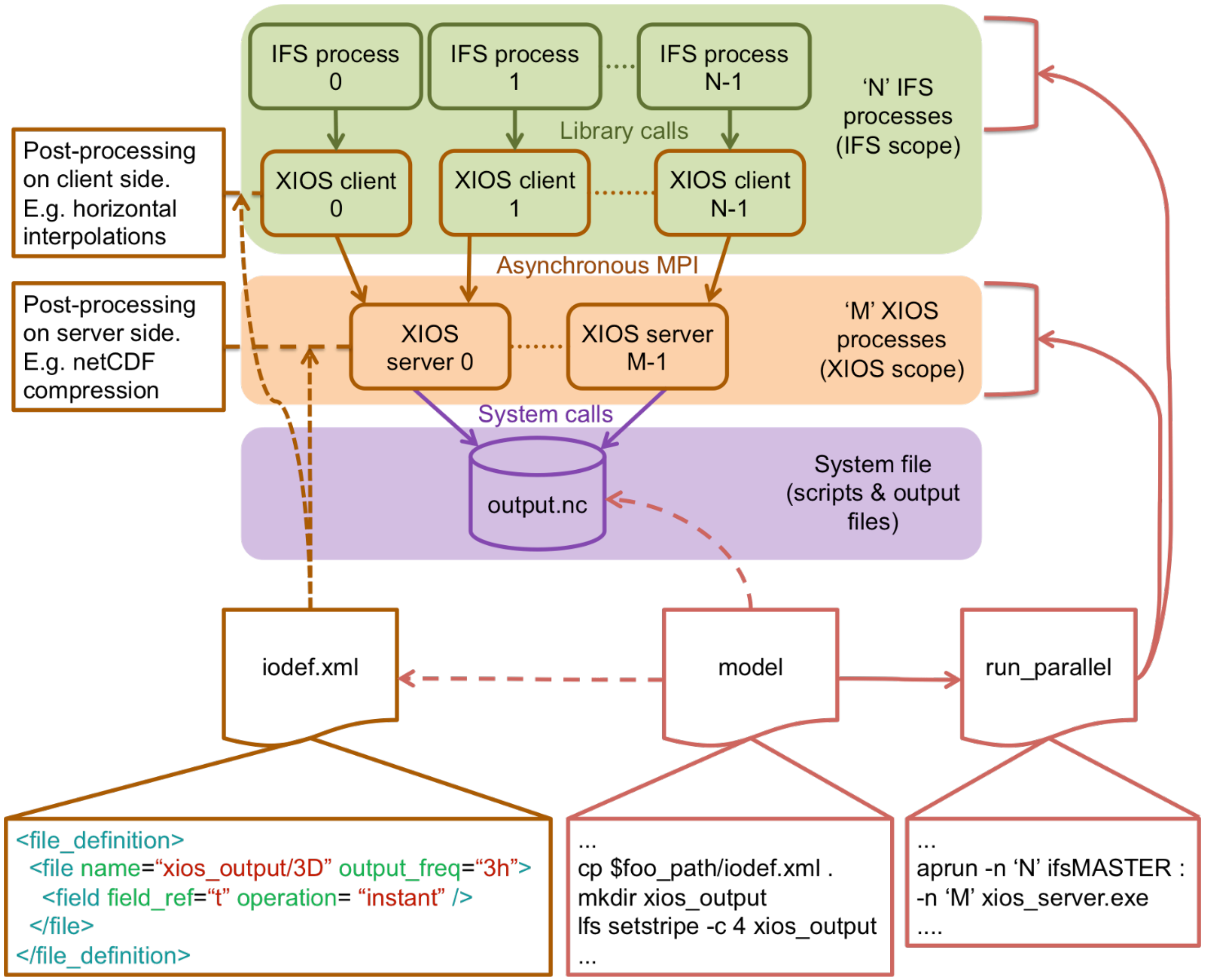
- The XML Input/Output Server (XIOS) is an asynchronous MPI parallel I/O server developed by IPSL
- It writes using the netCDF format
- Written data is CMIP-compliant (CMORized)
- It is able to post-process data online to generate diagnostics

3. IFS-XIOS integration

eScience center

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Scheme of the IFS-XIOS integration



Development steps

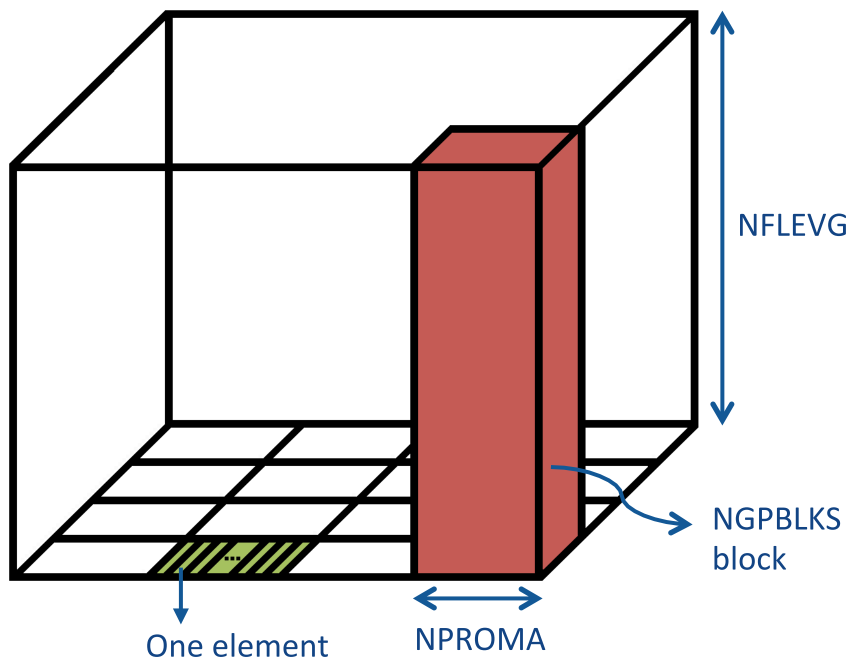
- XIOS setup
  - Initialization
  - Finalization
  - Context: calendar and geometry (axis, domain and grid)
  - *Iodef.xml* file
- Grid-point fields transfer
  - NPROMA blocks gather
  - Send fields
- Environment setup
  - XIOS compilation
  - Include and link XIOS, netCDF and HDF5
  - Model script
  - Supporting MPMD mode
- FullPos integration to support vertical post-processing: grid-point fields only

Development steps

- XIOS setup
  - Initialization
  - Finalization
  - Context: calendar and geometry (axis, domain and grid)
  - *Iodef.xml* file
- Grid-point fields transfer
  - **NPROMA blocks gather**
  - Send fields
- Environment setup
  - XIOS compilation
  - Include and link XIOS, netCDF and HDF5
  - Model script
  - Supporting MPMD mode
- **FullPos integration** to support vertical post-processing: grid-point fields only

Subdomain decomposition in IFS

- IFS uses a blocking strategy to efficiently parallelize the manipulation of data arrays using OpenMP
- IFS_data_array(NPROMA, NFLEVG, NFIELDS, NGPBLKS)



NPROMA blocks gather

- The IFS data arrays do not match with the XIOS ones:
  - IFS_data_array(NPROMA, NFLEVG, NFIELDS, NGPBLKS)
  - XIOS_data_array(unidimensional 2D domain, NFLEVG)
- We have to re-shuffle fields data before sending them
- According to the blocking strategy used in IFS, we have to build an XIOS-style array by gathering NPROMA blocks

FullPos integration

- FullPos is a post-processing package currently used by IFS
- The use of FullPos is necessary to perform vertical interpolations not supported by XIOS
- It is called as usual, and afterwards, data is sent to XIOS
- For now, it is only possible to output grid-point fields

4. Performance analysis and optimization

eScience center

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Execution overview

- Two different experiments according to if we use FullPos or not:
  - *Pre-FullPos*: it outputs both grid-point and spectral fields. NetCDF files size: 3.2 TB
  - *Post-FullPos*: it outputs only grid-point fields. NetCDF files size: 2.5 TB
- For both experiments:
  - Octahedral reduced Gaussian grid. Horizontal resolution: T1279 (16 km)
  - 702 MPI processes, each with 6 OpenMP threads
  - 10 days of forecast with a time step of 600 seconds
- Three optimizations applied:
  - Parallelization using OpenMP threads (both exp.) [XIOS v2]
  - Optimized compilation of XIOS with *-O3* (both exp.) [XIOS v3]
  - Computation and communication overlap (only *Pre-FullPos*)

Execution overview

- Execution times of this presentation correspond to *Post-FullPos*, except the third optimization
- Execution times:
  - Sequential output: 9391 seconds
  - MF I/O server: 7453 seconds
  - IFS-XIOS integration: 7682 seconds [XIOS v1]
  - IFS with FullPos but no I/O: 7443 seconds

Threading with OpenMP

- We used OpenMP to parallelize the NPROMA blocks gather
- Although it does not have a big impact, it avoids the rest of threads to be idle while the master is working
- The execution time is reduced 80 seconds, from 7682 seconds to 7602 seconds

Optimized compilation of XIOS

- We had a lot of issues to optimally compile XIOS
- For this reason, we used a conservative option: *-O1*
- XIOS reports too much time for just outputting data:

Client

```
-> report :  Performance report : Whole time from XIOS init and
   ↪ finalize: 7681.68 s
-> report :  Performance report : total time spent for XIOS :
   ↪ 132.715 s
-> report :  Performance report : time spent for waiting free
   ↪ buffer : 3.80519 s
-> report :  Performance report : Ratio : 0.0495359 %
```

Server

```
-> report : Performance report : Time spent for XIOS : 7681.68
-> report : Performance report : Time spent in processing events :
   ↪ 3196.4
-> report : Performance report : Ratio : 41.6107%
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

eScience center

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Optimized compilation of XIOS

- Someone previously reported a bug in the compilation of XIOS using *-O2* and *-O3* for Cray compilers
- However, it was reported using older Cray compilers, so it might be solved in newer versions
- Certainly, XIOS compiled and tests successfully passed

Optimized compilation of XIOS

- The execution time in both client and server sides is reduced

Client

```
-> report :  Performance report : Whole time from XIOS init and
   ↪ finalize: 7562.36 s
-> report :  Performance report : total time spent for XIOS :
   ↪ 40.3018 s
-> report :  Performance report : time spent for waiting free
   ↪ buffer : 0.463693 s
-> report :  Performance report : Ratio : 0.00613159 %
```

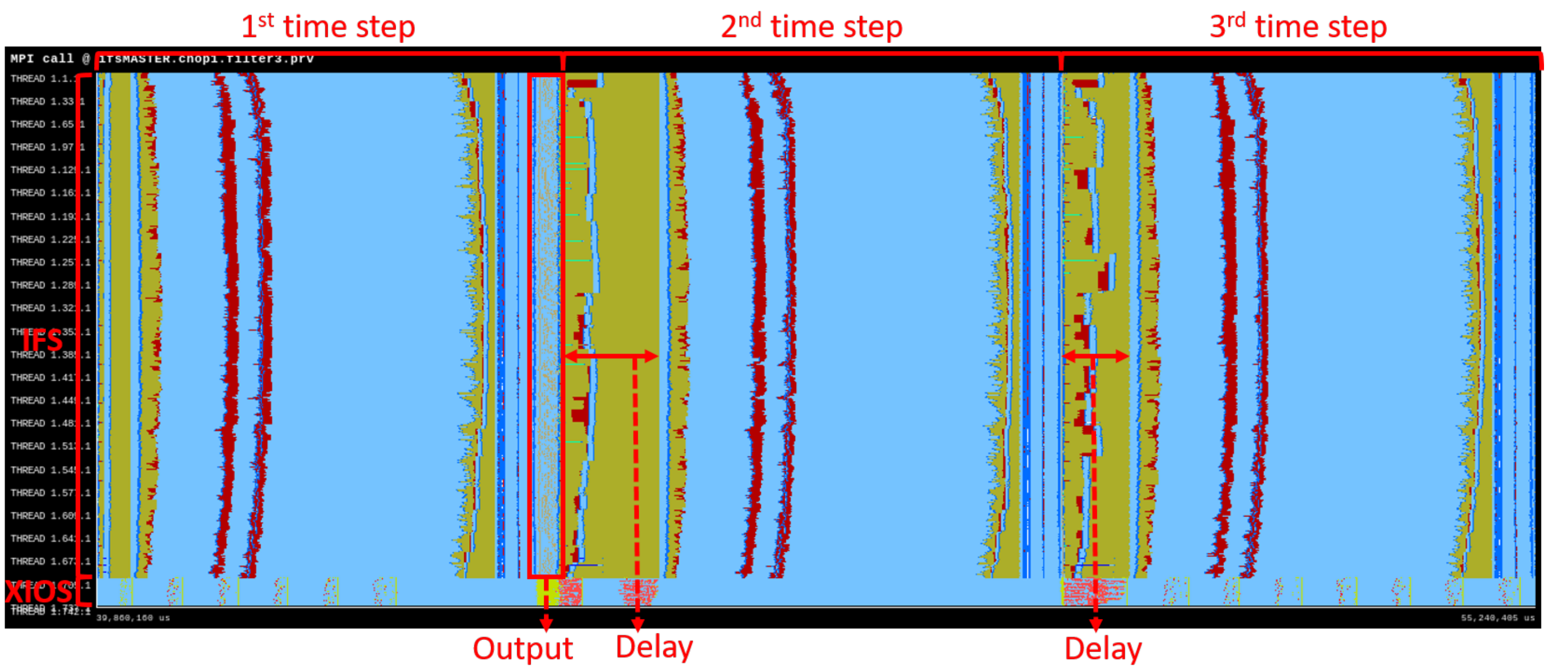
Server

```
-> report : Performance report : Time spent for XIOS : 7562.37
-> report : Performance report : Time spent in processing events :
   ↪ 1382.16
-> report : Performance report : Ratio : 18.2768%
```

- The execution time is reduced 103 seconds, from 7602 seconds to 7499 seconds

Overlapping computation and communication

- The trace shows that after an output time step, there is a delay in the communications of the next two time steps (*MPI_Waitany* and *MPI_Alltoallv*)
- There is a conflict between intra IFS communications and IFS to XIOS communications

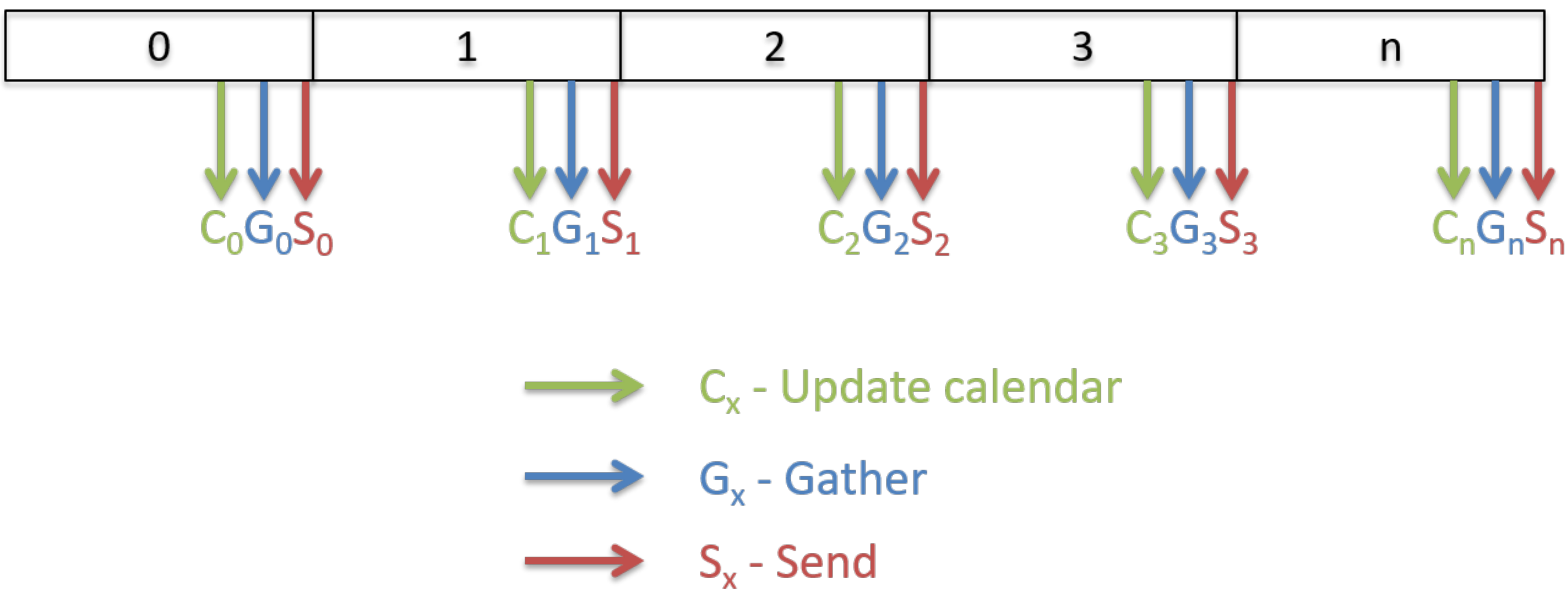


Overlapping computation and communication

Approach of the current output scheme:

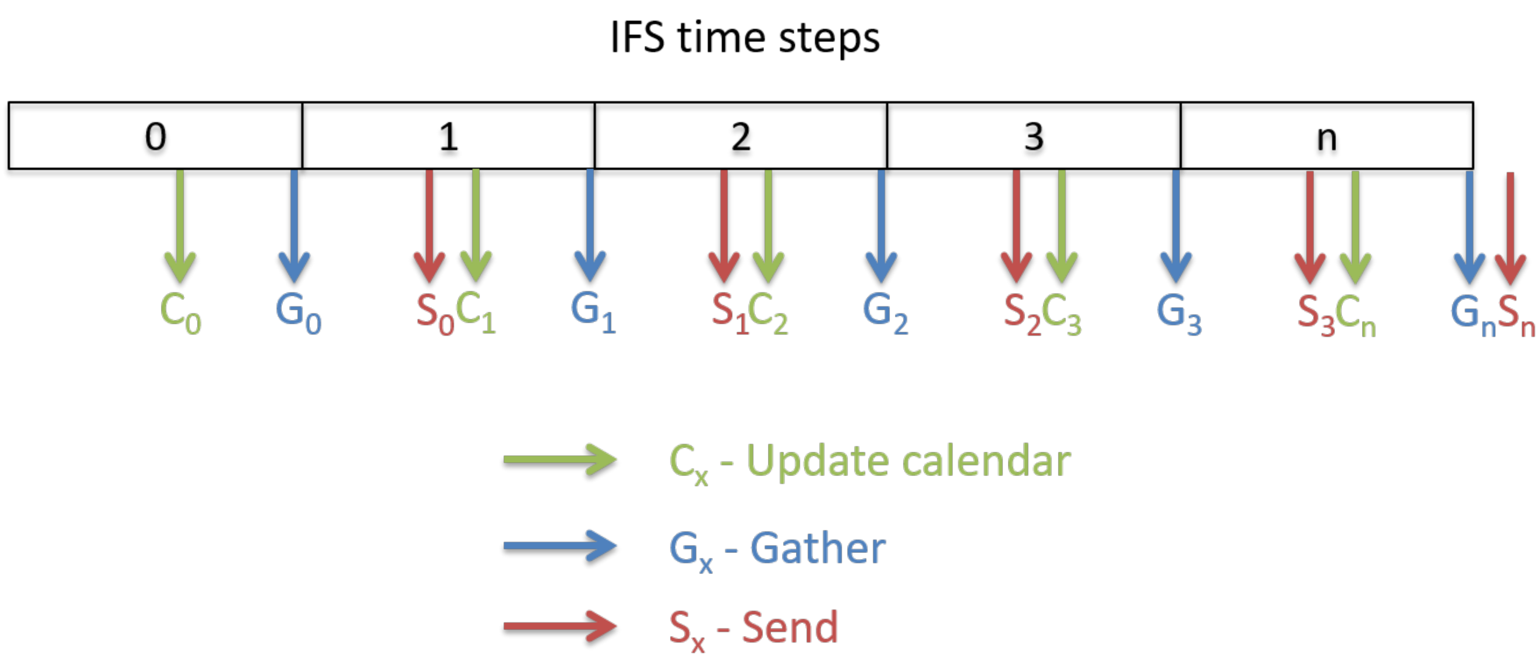
- If it is an output time step, at the end of it IFS sequentially executes three steps
- Otherwise, IFS only executes the update calendar step

IFS time steps



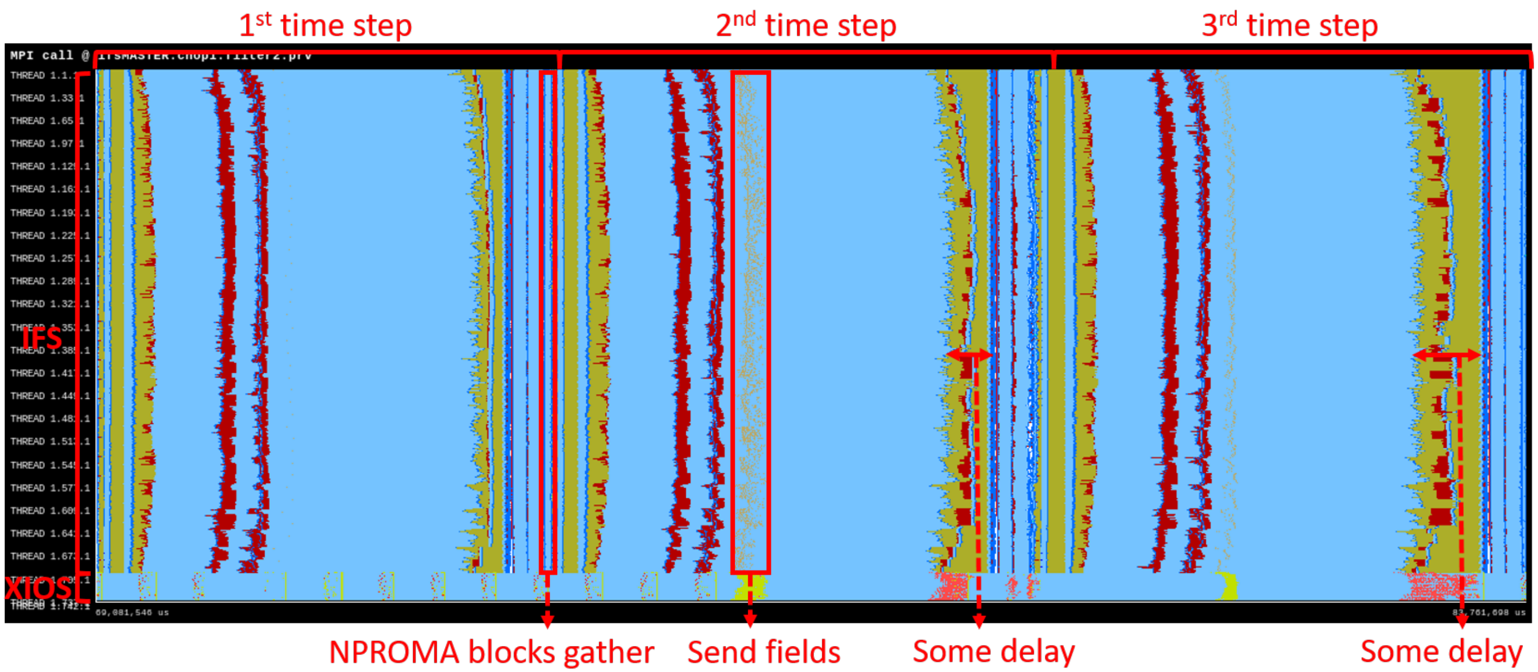
Overlapping computation and communication

- We used a new output scheme to truly overlap XIOS communication with IFS computation
- It splits the three needed steps to output data through XIOS:



Overlapping computation and communication

- The trace shows that there is no delay at the beginning of the 2nd and 3rd time steps. However, there is some delay at the end, but it is less significant
- The execution time is reduced 122 seconds (*Pre-FullPos)*

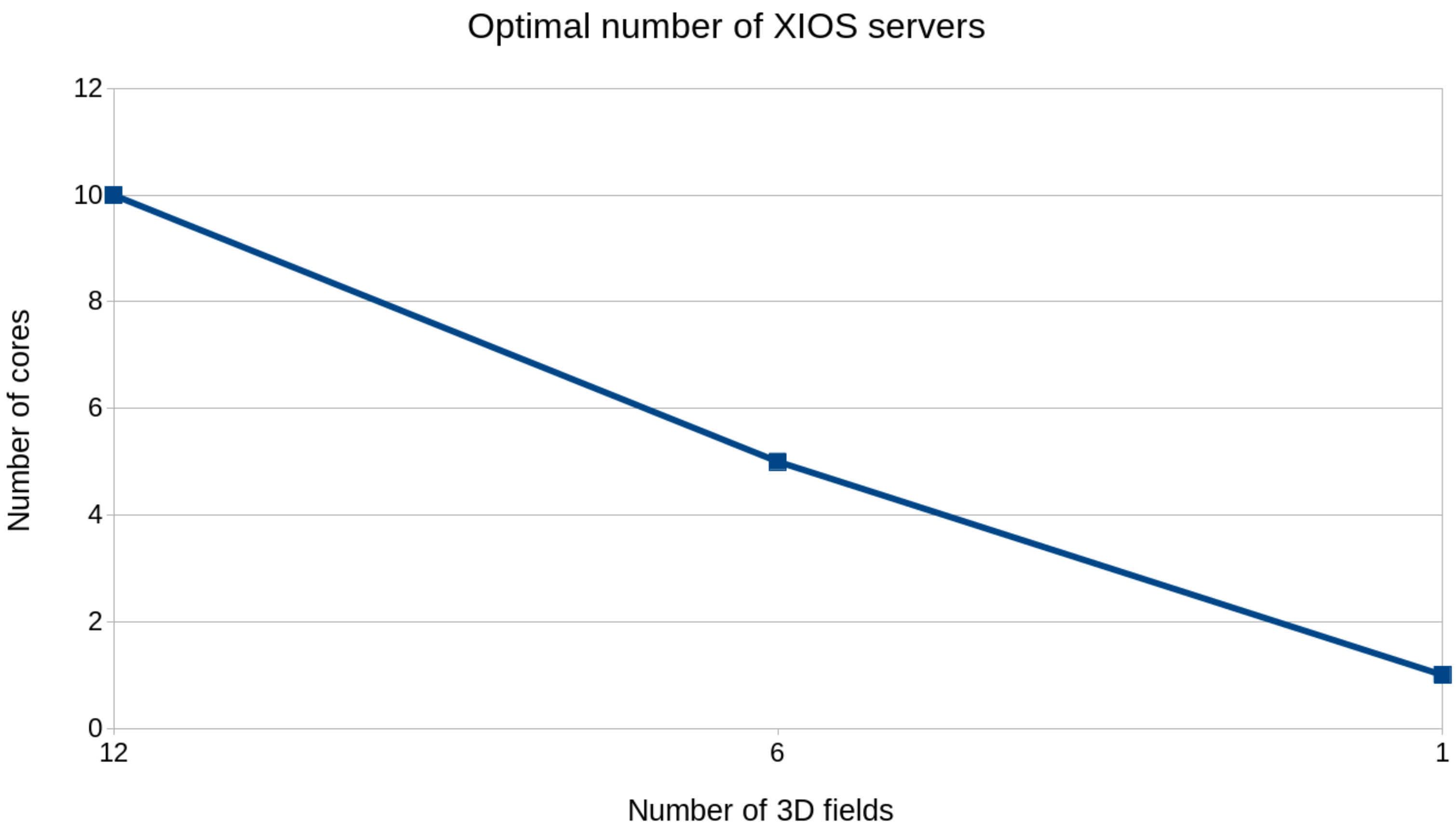


5. Evaluation

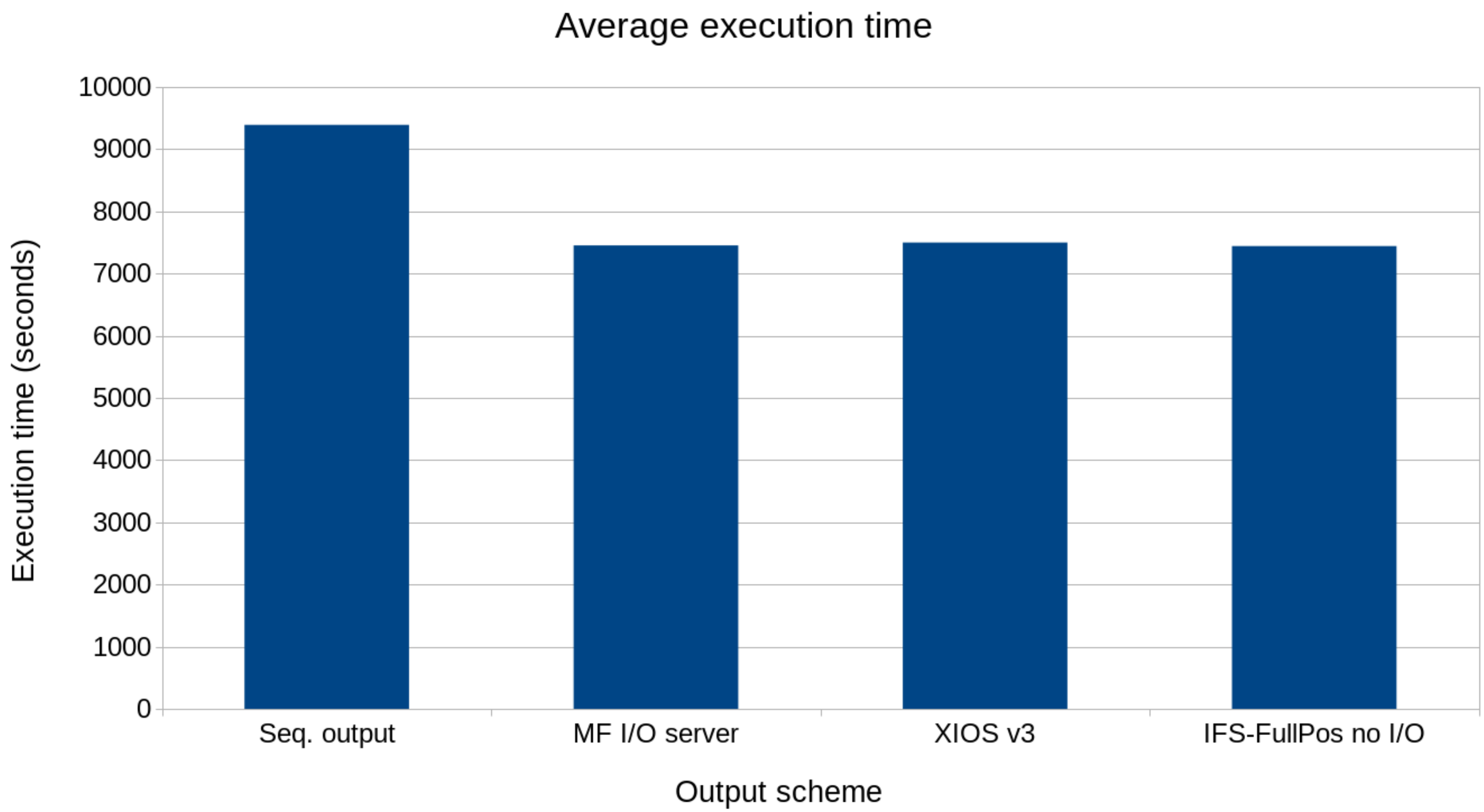
eScience center

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

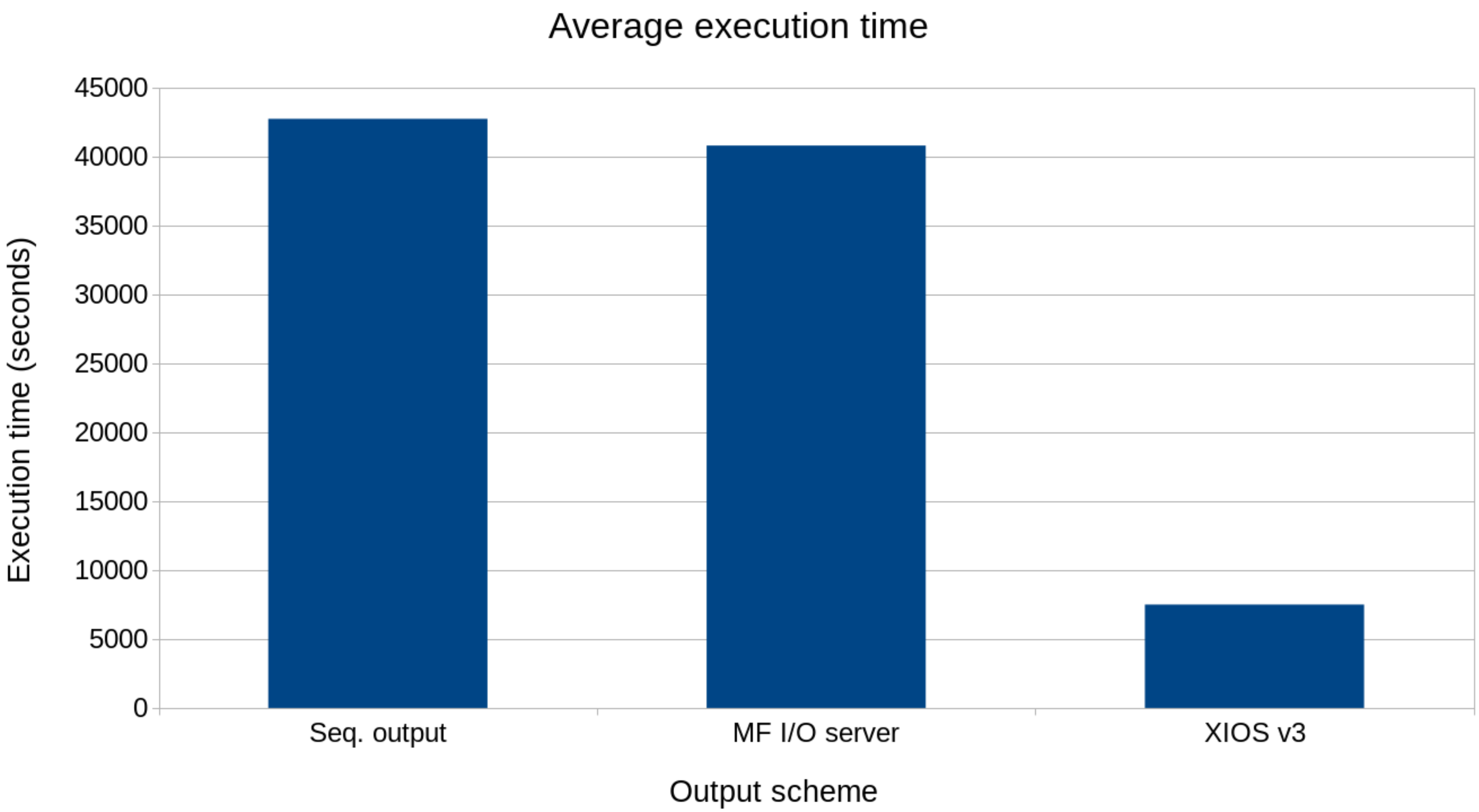
Optimal number of XIOS servers



Comparison test



Comparison test adding GRIB to netCDF post-processing



6. Conclusions

eScience center

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Conclusions

- We have presented an easy-to-use development
- The integration with no optimization already improved the execution time:
  - Sequential output 9391 seconds (20.7% of overhead) → IFS-XIOS integration 7682 seconds (3.1% of overhead)
- Using OpenMP to parallelize the NPROMA blocks gather, the execution time is reduced by 80 seconds
  - It is really important to make an scalable gather, because it could become a bottleneck for future higher grid resolutions

Conclusions

- Using an optimized compilation of XIOS, the execution time is reduced by 103 seconds
  - This optimization proves that it is important to compile external libraries using the best optimization flags
- Using a better overlapping between IFS computation and XIOS communication, the execution time is reduced by 122 seconds *(Pre-FullPos* experiment)
  - It is sometimes necessary to analyse in which places computation and communication can be effectively overlapped

Conclusions

- Performance highlights of the most optimized version:
  - It only has a 0.7% of overhead
  - Within 56 seconds IFS outputs 2.5 TB of data
- When post-processing to convert GRIB to netCDF files is taken into account:
  - The post-processing takes 9.2 hours (sequentially performed, as in EC-Earth)
  - Thus, the most optimized version is a 5.7x faster than the sequential output and a 5.4x faster than the MF I/O server

Conclusions

- These numbers denote that we have implemented an scalable and efficient development that will address the I/O issue
- In EC-Earth, this new I/O development will:
  - Increase the performance and efficiency of the whole model
  - Perform online post-processing operations
  - Save thousands of computing hours
  - Save storage space, because it will only store processed data ready to be used

Conclusions

- These numbers denote that we have implemented an scalable and efficient development that will address the I/O issue
- In EC-Earth, this new I/O development will:
  - Increase the performance and efficiency of the whole model
  - Perform online post-processing operations
  - Save thousands of computing hours
  - Save storage space, because it will only store processed data ready to be used

Save money!

Ongoing and future work

- Use FullPos to transform fields from spectral space to grid-point space, post-process and send them to XIOS
- The development done for IFS will be ported to OpenIFS
- Adapt the EC-Earth model to generate online diagnostics from OpenIFS and NEMO components through XIOS

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

eScience center

Thank you!

xavier.yepes@bsc.es

Computational aspects and performance evaluation of the IFS-XIOS integration

Xavier Yepes-Arbós, BSC
Mario C. Acosta, BSC
Gijs van den Oord, NLeSC
Glenn Carver, ECMWF

18th workshop on high performance computing in meteorology, ECMWF, UK

24/09/2018