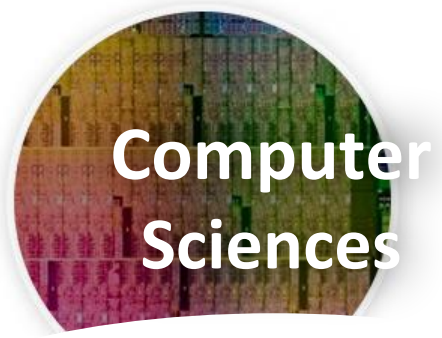

Barcelona Supercomputing Center
Centro Nacional de Supercomputación


# Performance activities for Earth System Modelling

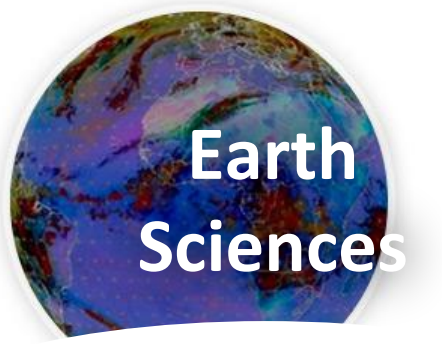
**Earth Science Department (BSC)**
**Computational Earth Science**
Performance Team

Mario C. Acosta, Miguel Castrillo, Oriol Tintó, Xavier Yepes

# BSC Departments

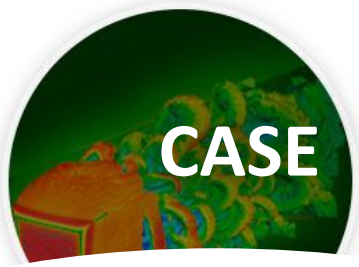



To influence the way machines are built, programmed and used: programming models, performance tools, Big Data, computer architecture, energy efficiency




To develop and implement global and regional state-of-the-art models for short-term air quality forecast and long-term climate applications


To understand living organisms by means of theoretical and computational methods (molecular modeling, genomics, proteomics)



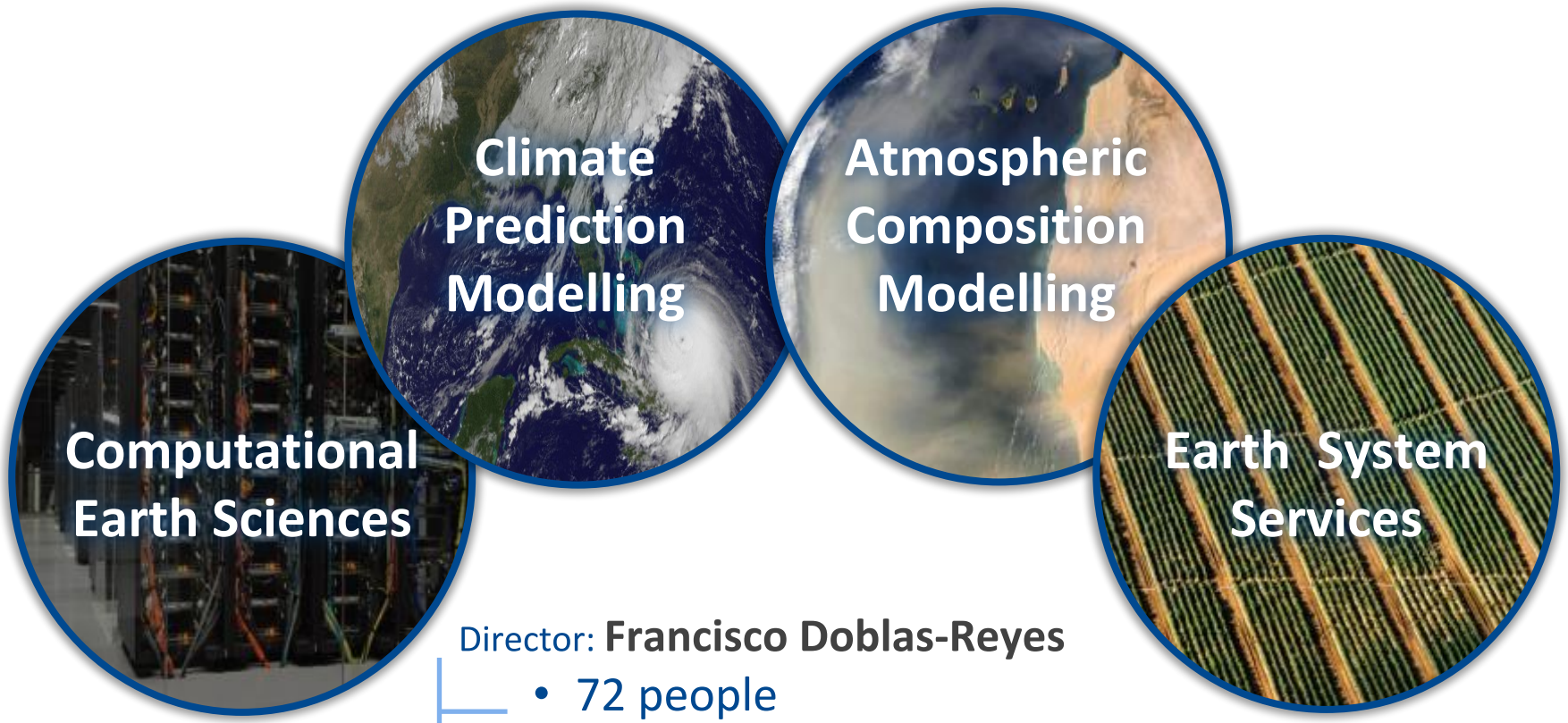

To develop scientific and engineering software to efficiently exploit super-computing capabilities (biomedical, geophysics, atmospheric, energy, social and economic simulations)

# Earth Science Department

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Environmental modelling and forecasting, with a particular focus on weather, climate and air quality

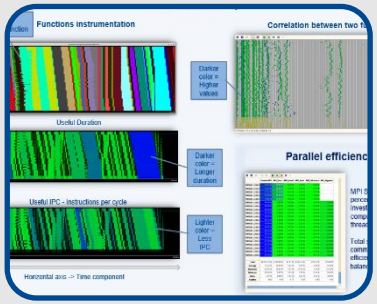



Director: **Francisco Doblas-Reyes**

- 72 people
- Leading: H2020 project, COPERNICUS contract, ERC Consolidator Grant and hosts an AXA Chair

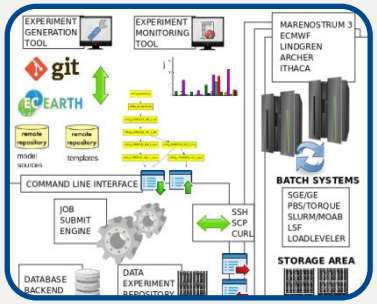
# Computational Earth Science





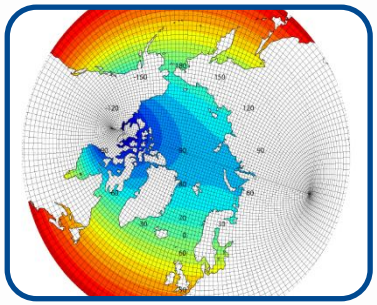

## Performance Team

- Provide HPC Services such as performance analysis or optimizations for Earth System Models
- Research on new computational methods




## Models and Workflows Team

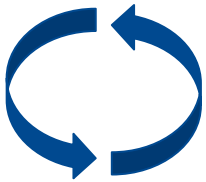
- Development of HPC user-friendly software framework
- Support the development of atmospheric research software



## Data and Diagnostics Team

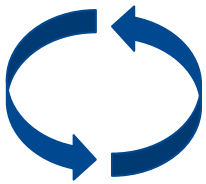
- Big Data in Earth Sciences
- Provision of data services
- Visualization

**Weather and Climate Science**



**High Performance Computing (Services and Research) applied to Earth System Modelling**

- Knowledge about the mathematical and computational side of Earth System Applications
- Knowledge about the specific needs in HPC of the Earth System Applications
- Researching about HPC methods specifically used for Earth System Applications



**Computer Science**

- Future H2020 projects and proposals where we work/will work
  - ESCAPE2 → Profiling analysis during benchmarking
  - MERCATOR → Profiling analysis and research in new optimizations for NEMO
  - ESiWACE2 → EC-Earth 5km, HPC services for pre-exascale, Efficient IFS/XIOS integration
  - IS-ENES3 → Co-leading HPC workpackage
  - COPERNICUS → Profiling analysis and research in new optimizations for NEMO
  - HARMONIE-AROME → Profiling analysis and research in new optimizations (Proposal to Hirlam advisory Committee)
  - XIOS → Profiling analysis and research in new optimizations (Collaboration with XIOS Team, IPSL)

# Methodology

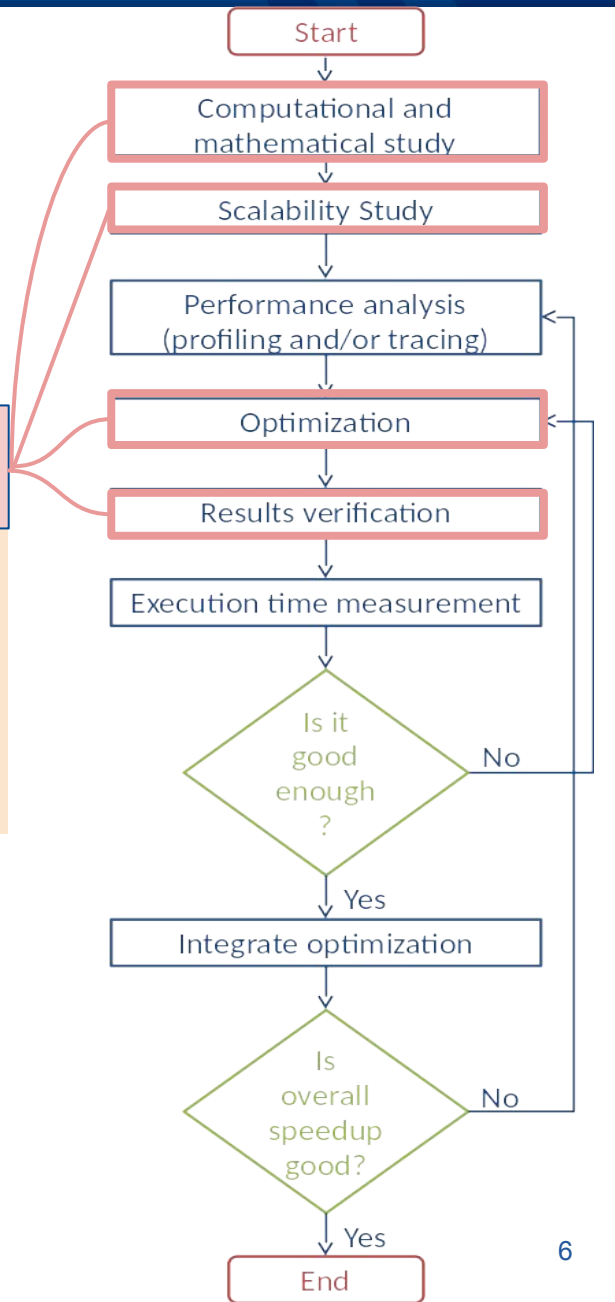


**Weather and Climate Science**

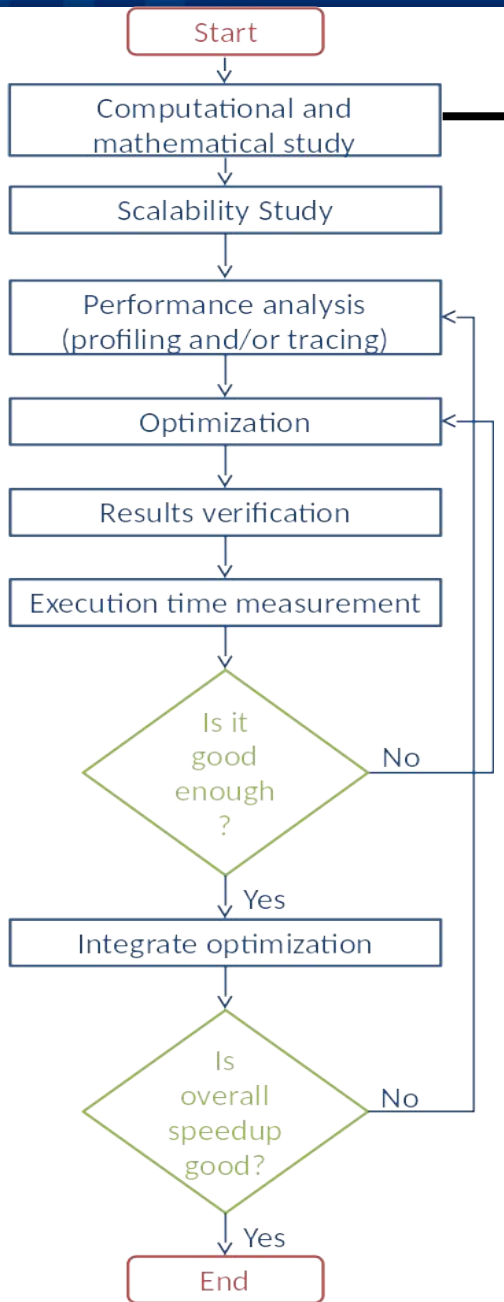
**High Performance Computing (Services and Research) applied to Earth System Modelling**

- Knowledge about the mathematical and computational side of Earth System Applications
- Knowledge about the specific needs in HPC of the Earth System Applications
- Researching about HPC methods specifically used for Earth System Applications

**Computer Science**

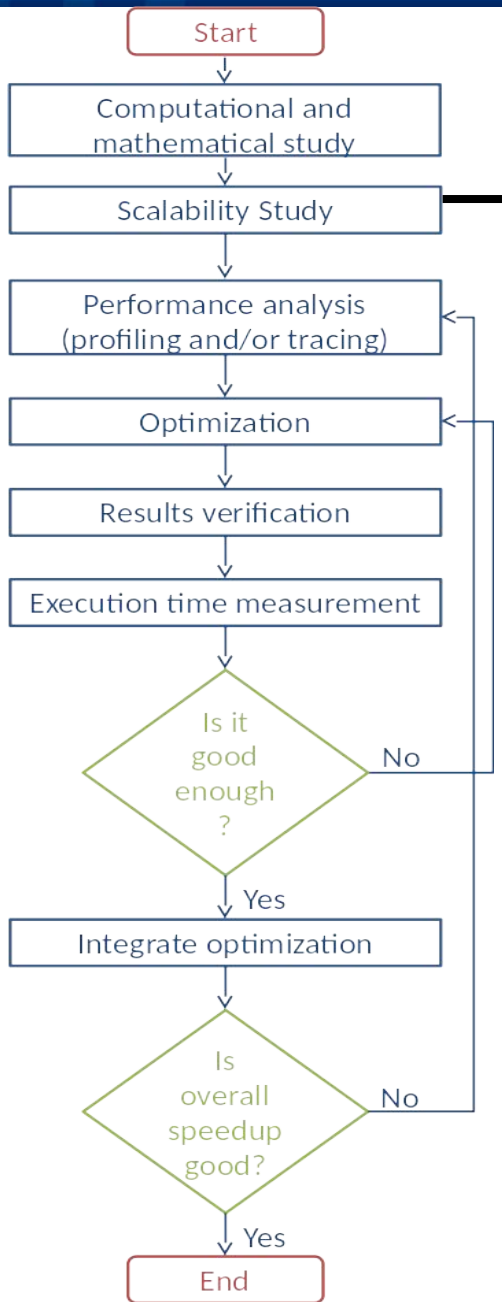


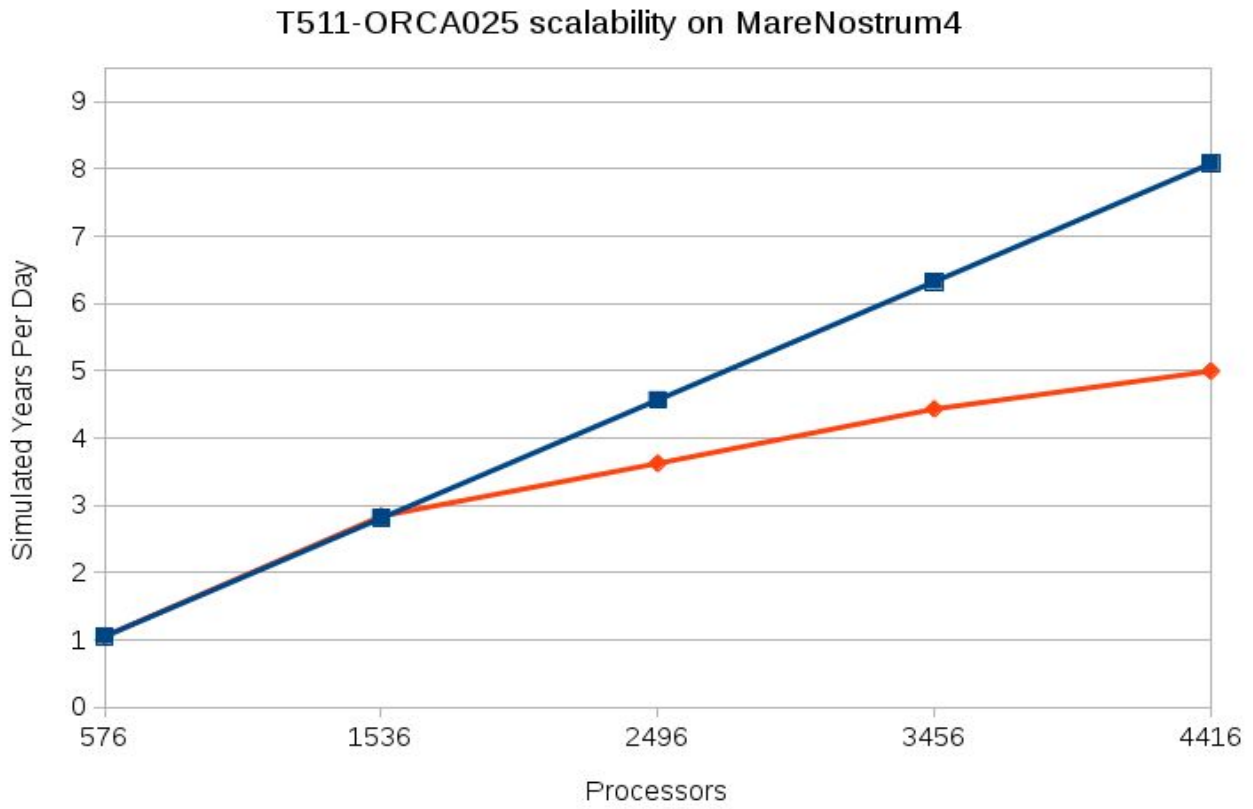
# Methodology

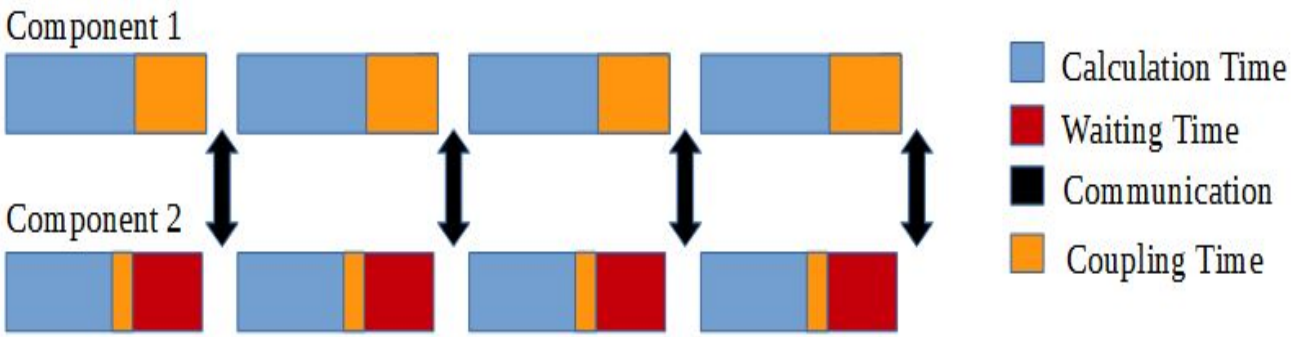



- Mathematical study
  - Some methods could be better than others
    - Discretization used (explicit, implicit, semi-implicit…)
    - Parallel adaptation (solvers, preconditioners…)
  - How to implement new algorithms for new architectures
- Computational study
  - Achieve load balance among components
  - Reduce overhead introduced by parallel applications
  - Assure that the computational algorithm takes advantage of the architecture

# Methodology

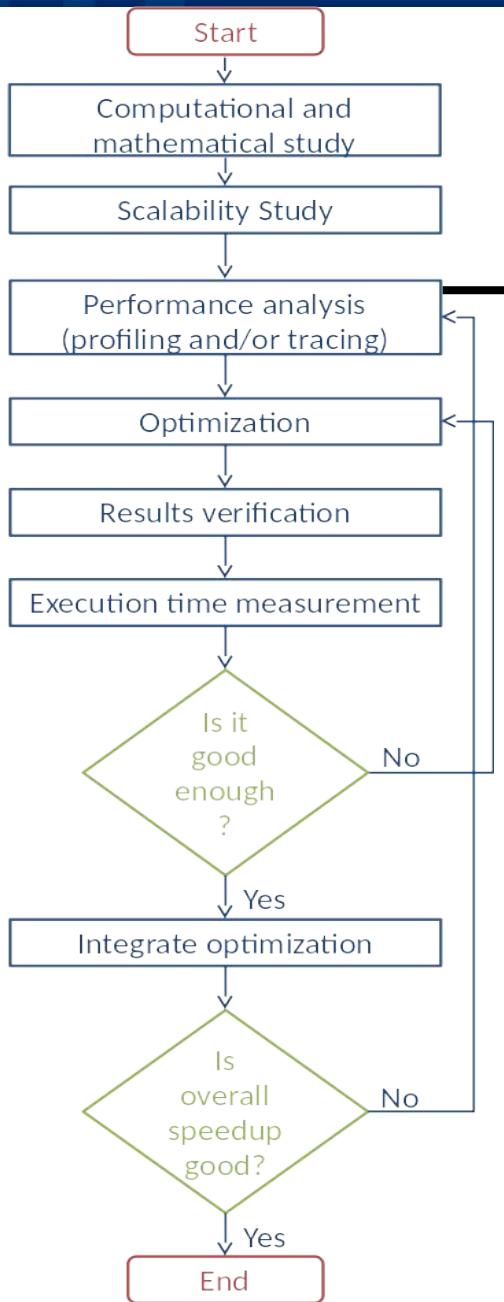


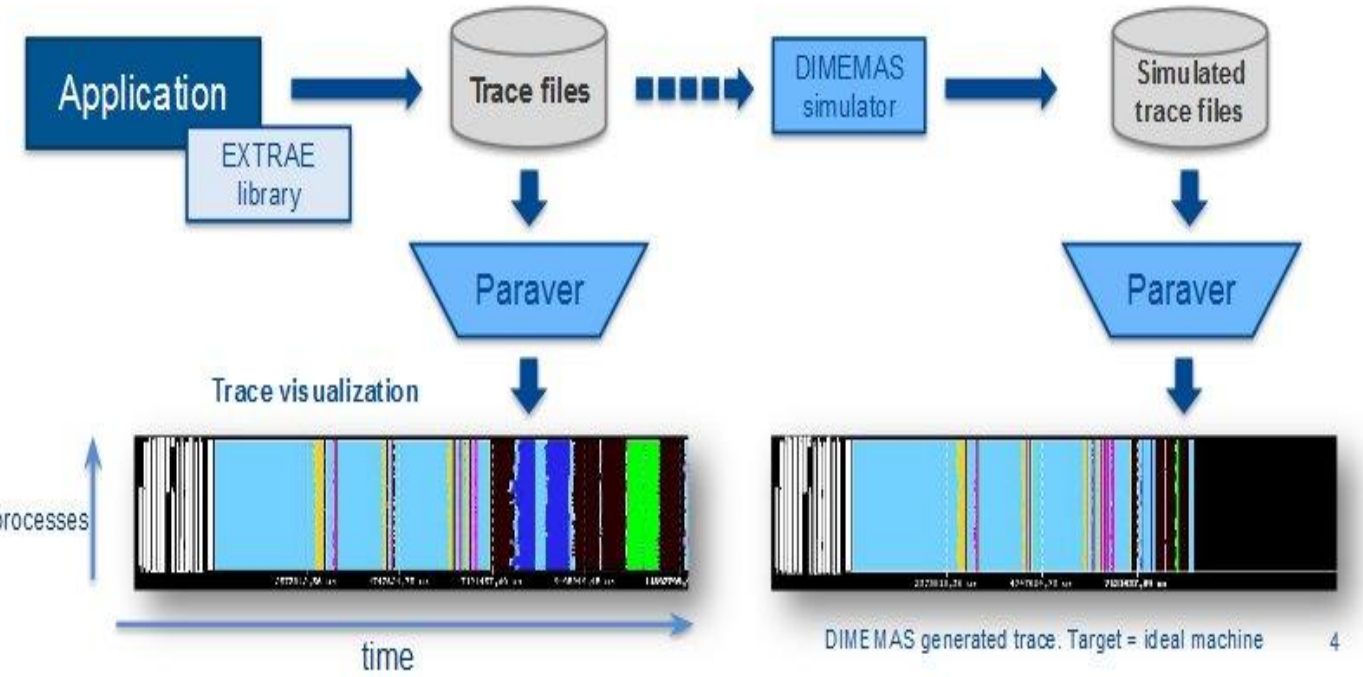




Possible load balance of coupled components of a Earth System Model

# Methodology




- Since 1991
- Based on traces
- Open Source: http://www.bsc.es/paraver
- **Extrae**: Package that generates Paraver trace-files for a post-mortem analysis
- **Paraver**: Trace visualization and analysis browser
  - Includes trace manipulation: Filter, cut traces
- **Dimemas**: Message passing simulator

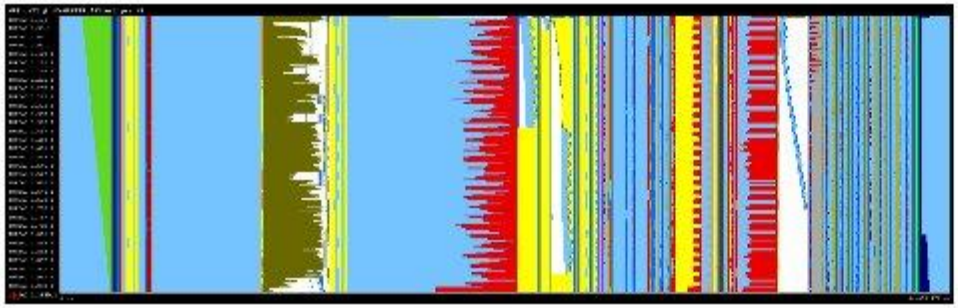


# General Profiling: Study Area

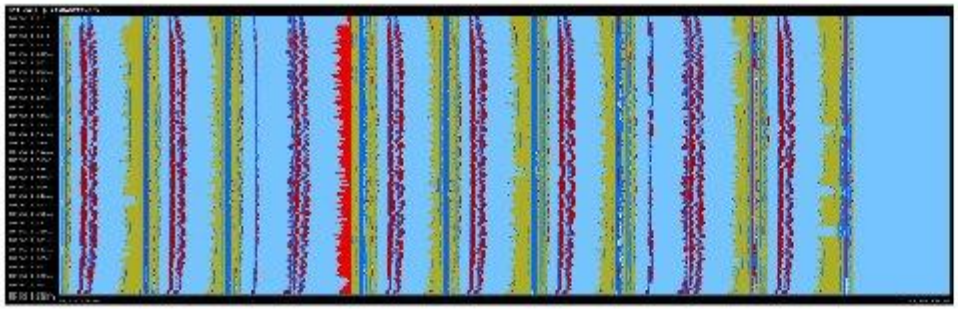


- General MPI profile+Histogram → localize your study area

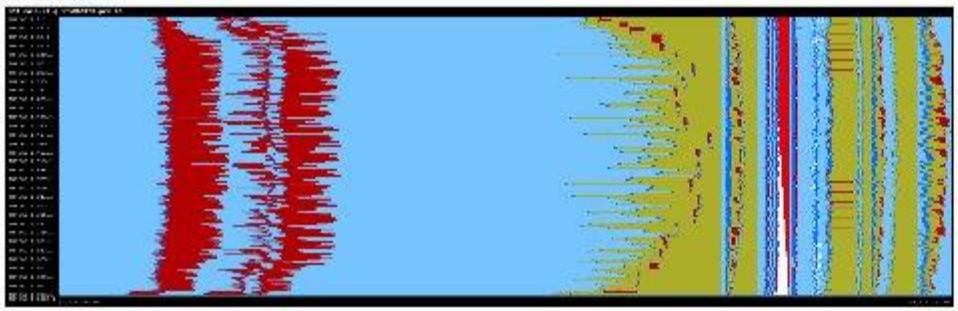
**IFS CY43R1 (T511)**



→ Complete execution



→ Some time steps



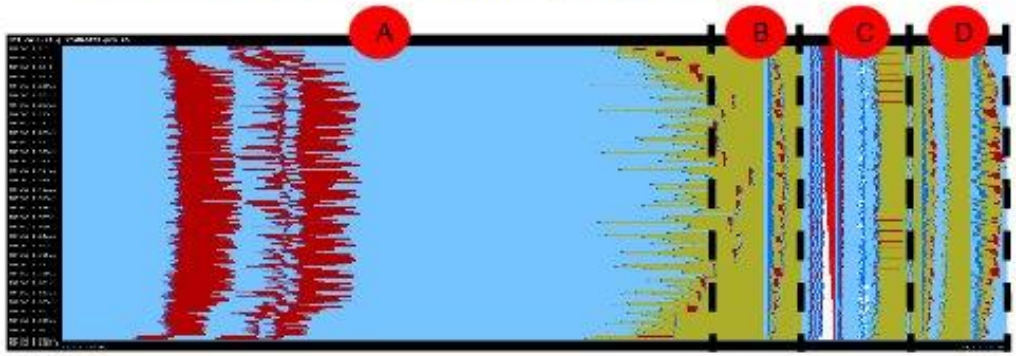
→ One time step

48

# General Profiling



- Localize each scientific phase in your model and evaluate it independently

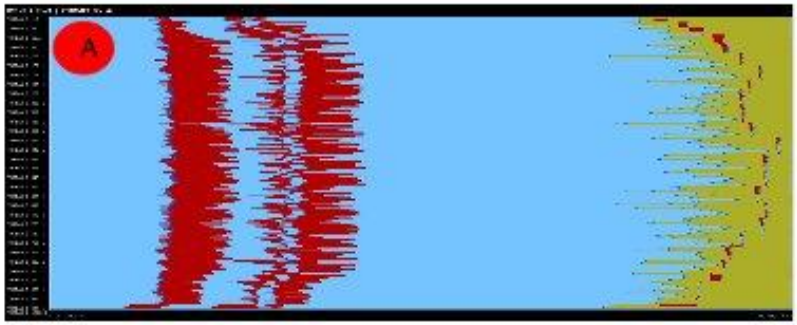


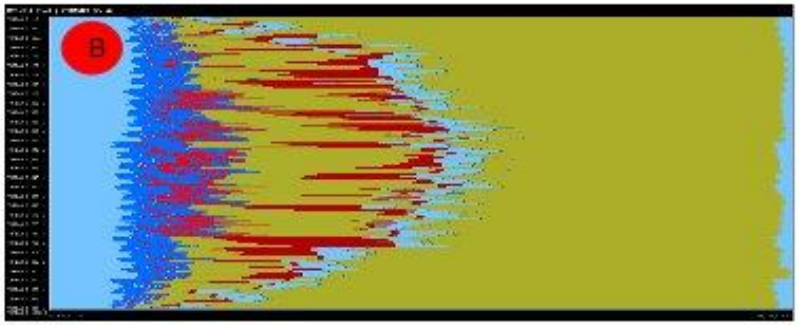

A Grid Point Calculations

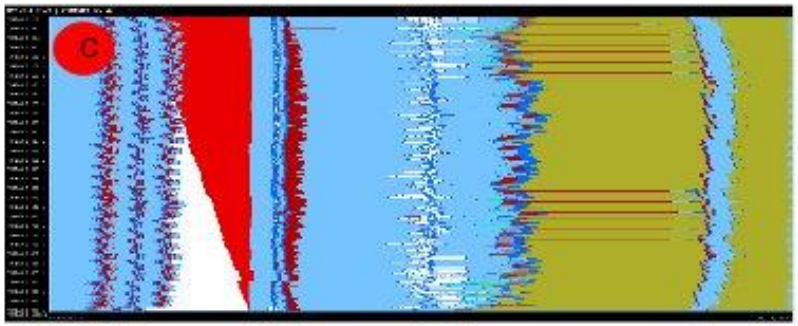
B Transformations and Transpositions (Fourier + Legendre)

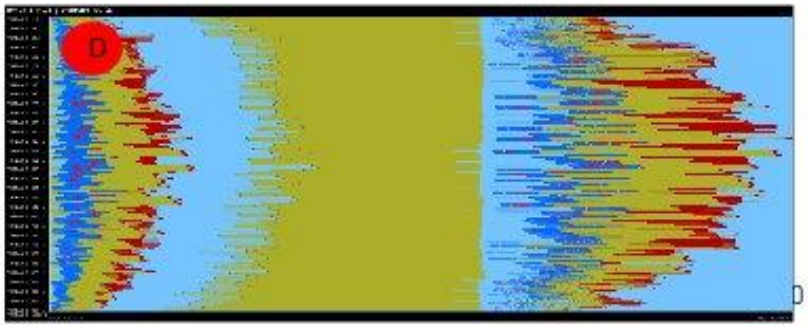
C Spectral Calculations

D Fourier + Legendre Inverse



# General Profiling

- Parallel and Communication efficiency, Global load balance → less than 85%?

Parallel Efficiency

IFS standalone

| | Outside MPI | MPI_Send | MPI_Recv | MPI_Isend | MPI_Irecv | MPI_Wait | MPI_Barrier | MPI_Alltoallv | MPI_Gatherv | MPI_Comm_rank | MPI_Comm_size | MPI_Bsend | MPI_Waitany |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Total | 66,578.44 % | 1.71 % | 773.76 % | 646.21 % | 239.35 % | 12,362.37 % | 806.93 % | 10,757.31 % | 35.56 % | 2.49 % | 448.23 % | 0.81 % | 7,746.82 % |
| Average | 66.31 % | 0.00 % | 0.77 % | 0.64 % | 0.24 % | 12.31 % | 0.80 % | 10.71 % | 0.04 % | 0.00 % | 0.45 % | 0.81 % | 7.72 % |
| Maximum | 72.93 % | 0.01 % | 2.98 % | 1.60 % | 0.80 % | 18.56 % | 1.84 % | 25.06 % | 1.12 % | 0.01 % | 1.88 % | 0.81 % | 19.25 % |
| Minimum | 57.05 % | 0.00 % | 0.01 % | 0.08 % | 0.07 % | 3.11 % | 0.00 % | 5.25 % | 0.00 % | 0.00 % | 0.16 % | 0.81 % | 0.31 % |
| StDev | 2.03 % | 0.00 % | 0.57 % | 0.36 % | 0.06 % | 2.52 % | 0.41 % | 3.57 % | 0.12 % | 0.00 % | 0.10 % | 0 % | 3.18 % |
| Avg/Max | 0.91 | 0.31 | 0.26 | 0.40 | 0.30 | 0.66 | 0.44 | 0.43 | 0.03 | 0.34 | 0.24 | 1 | 0.40 |

MPI_Send

Communication Efficiency

Global Load Balance

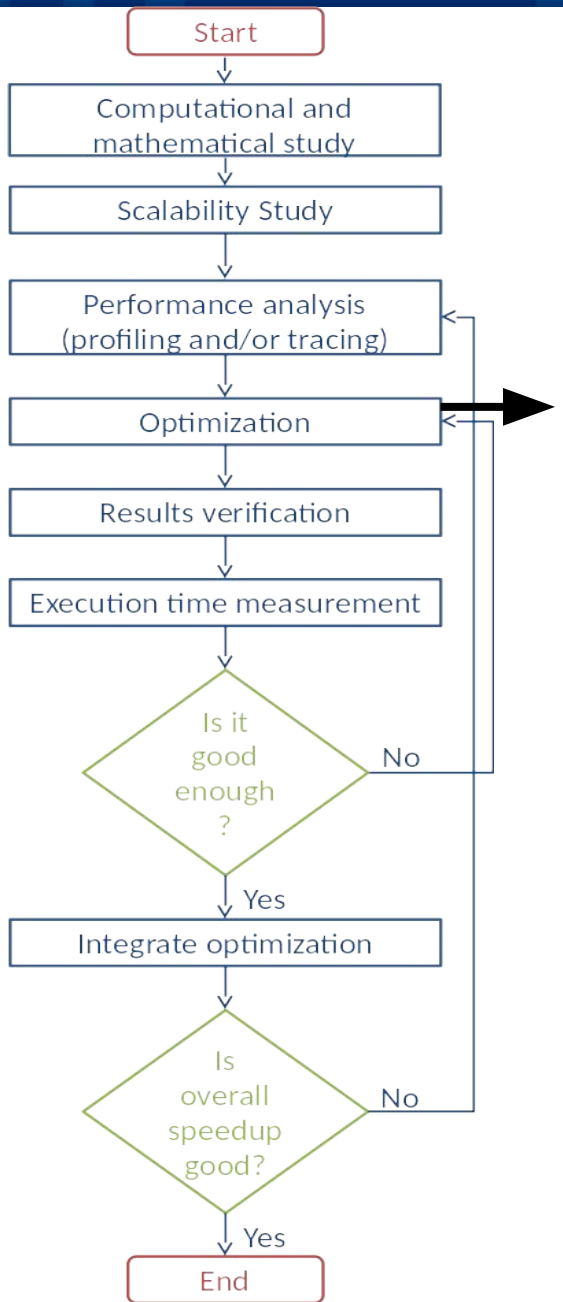
NEMO+Coupling

MPI call profile @ ifsMASTER.chop2.prv

| | Outside MPI | MPI_Send | MPI_Recv | MPI_Isend | MPI_Irecv | MPI_Wait | MPI_Allreduce | MPI_Alltoallv | MPI_Gatherv | MPI_Comm_rank | MPI_Comm_size | MPI_Waitany |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Total | 15,664.28 % | 0.27 % | 1,870.84 % | 190.00 % | 3.58 % | 940.53 % | 2,591.31 % | 282.28 % | 6.17 % | 0.17 % | 7.15 % | 43.42 % |
| Average | 72.52 % | 0.00 % | 8.66 % | 0.88 % | 0.02 % | 4.35 % | 12.00 % | 1.31 % | 0.03 % | 0.00 % | 0.03 % | 0.20 % |
| Maximum | 74.16 % | 0.00 % | 12.71 % | 1.10 % | 0.04 % | 9.07 % | 13.45 % | 1.42 % | 0.05 % | 0.00 % | 0.05 % | 0.60 % |
| Minimum | 70.16 % | 0.00 % | 3.80 % | 0.62 % | 0.01 % | 1.01 % | 10.88 % | 1.18 % | 0.00 % | 0.00 % | 0.01 % | 0.05 % |
| StDev | 0.79 % | 0.00 % | 2.39 % | 0.08 % | 0.00 % | 2.43 % | 0.46 % | 0.05 % | 0.02 % | 0.00 % | 0.01 % | 0.11 % |
| Avg/Max | 0.98 | 0.51 | 0.68 | 0.80 | 0.41 | 0.48 | 0.89 | 0.92 | 0.57 | 0.51 | 0.62 | 0.33 |

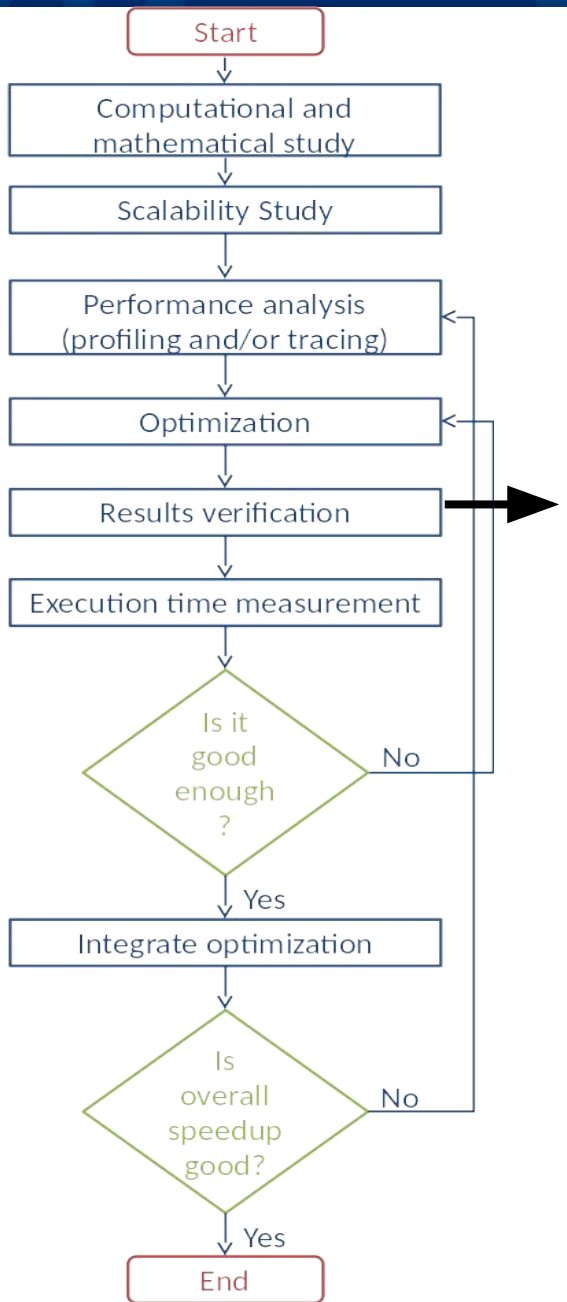
THREAD 1.3.1 MPI_Isend = 0.65 %

# Methodology

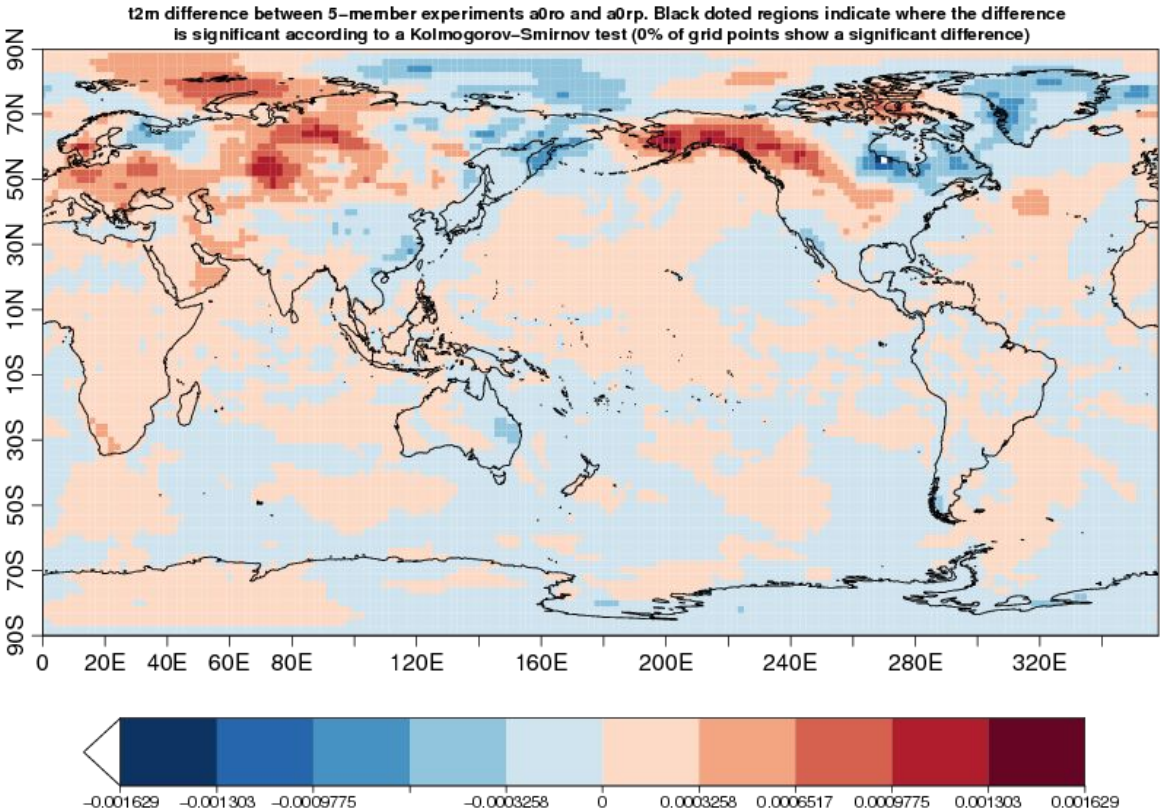



- Introducing optimizations
  - Improvement of the mathematical and/or computational algorithm
    - Apply scientific methods which are found in the literature
    - Improve the method with a new approach
  - Revolution: Create a new (and better) algorithm taking into account the research line followed

# Methodology




- Reproducibility study
  - Evaluate if the accuracy and reproducibility of the model is similar using or not the optimizations proposed
  - Take into account the nature of climate models
    - How to evaluate, in parallel executions, if the differences between runs are significant or not.



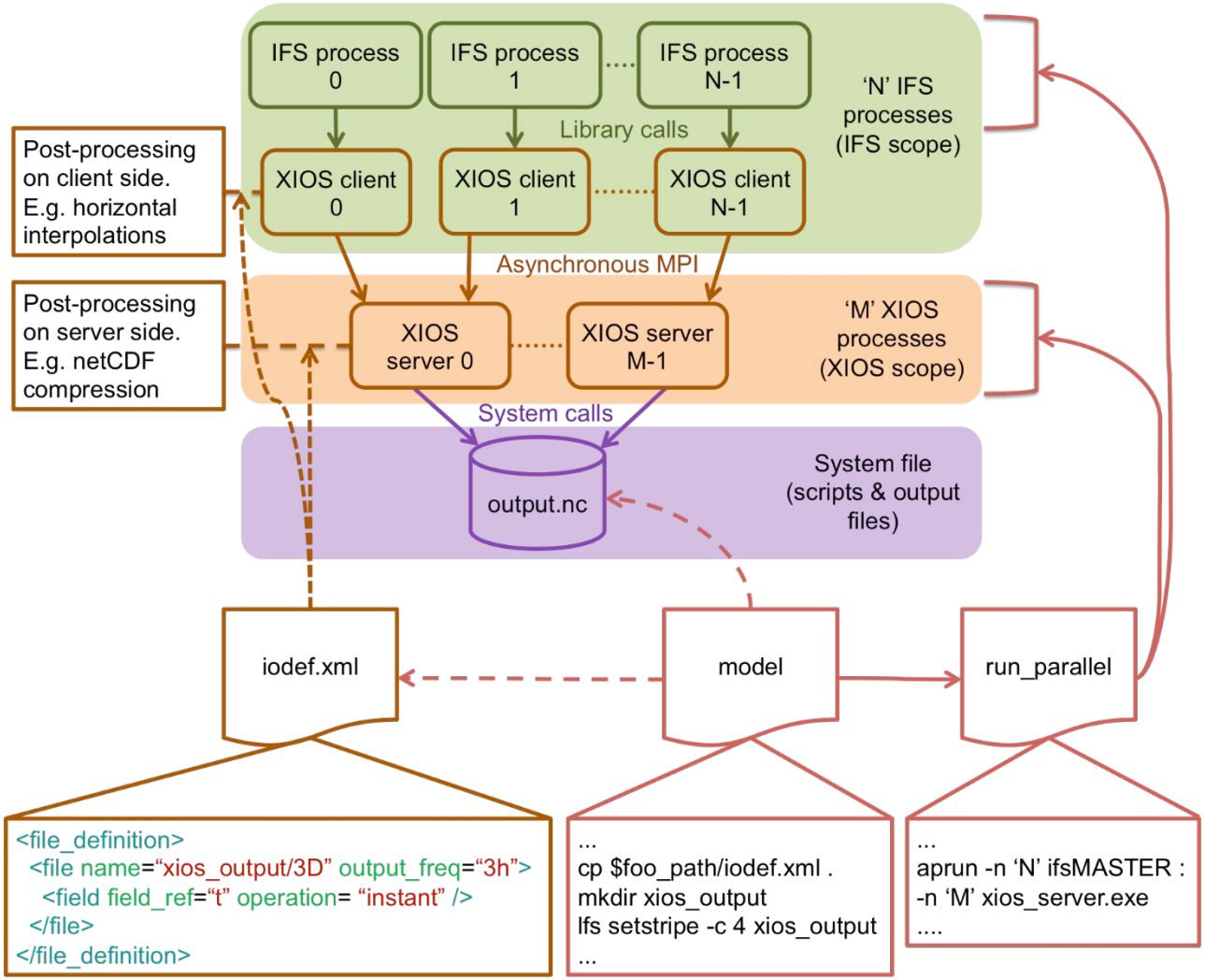
# IFS-XIOS Integration: Performance analysis and optimization

Xavier Yepes, Mario Acosta, Glenn Carver and Gijs van den Oord

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# Scheme of IFS-XIOS integration





- IFSCY43R3, T1259, 702 MPI processes, each with 6 OpenMP threads
- 10 days of forecast with a time step of 600 seconds
- Output size of netCDF files: 3.2 TB
- Execution times:
  - Sequential output: 9054 seconds
  - MF I/O server: 7535 seconds
  - IFS-XIOS integration: 7773 seconds
  - No output: 7356 seconds

# Threading with OpenMP



```
!$OMP PARALLEL PRIVATE(jstglo,icend,ibl,jlev)

! GFL - Specific humidity
IF (q) THEN
  !$OMP DO SCHEDULE(DYNAMIC)
  DO jstglo = 1, YDGEOMETRY%YRGEM%NGPTOT, YDGEOMETRY%YRDIM%
      ↪ NPROMA
    icend = MIN(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRGEM%
        ↪ NGPTOT-jstglo+1)
    ibl = (jstglo-1)/YDGEOMETRY%YRDIM%NPROMA + 1
    DO jlev = 1, YDGEOMETRY%YRDIMV%NFLEVG
      xios_gfl(jstglo:jstglo+icend-1,jlev) = YDFIELDS%YRGFL%GFL(1:icend,jlev,
          ↪ YGFL%YQ%MP,ibl)
    END DO
  END DO
  !$OMP END DO
  !$OMP SINGLE
  CALL xios_send_field("q",xios_gfl)
  !$OMP END SINGLE NOWAIT
END IF

! GMV - Temperature
IF (t) THEN
  !$OMP DO SCHEDULE(DYNAMIC)
  DO jstglo = 1, YDGEOMETRY%YRGEM%NGPTOT, YDGEOMETRY%YRDIM%
      ↪ NPROMA
```

Gather

Send

Gather and send overlap among two fields

# 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%
```

# Overlapping computation and communication



In an output time step, there is a slight increase in the execution time of the three following time steps

Non-output

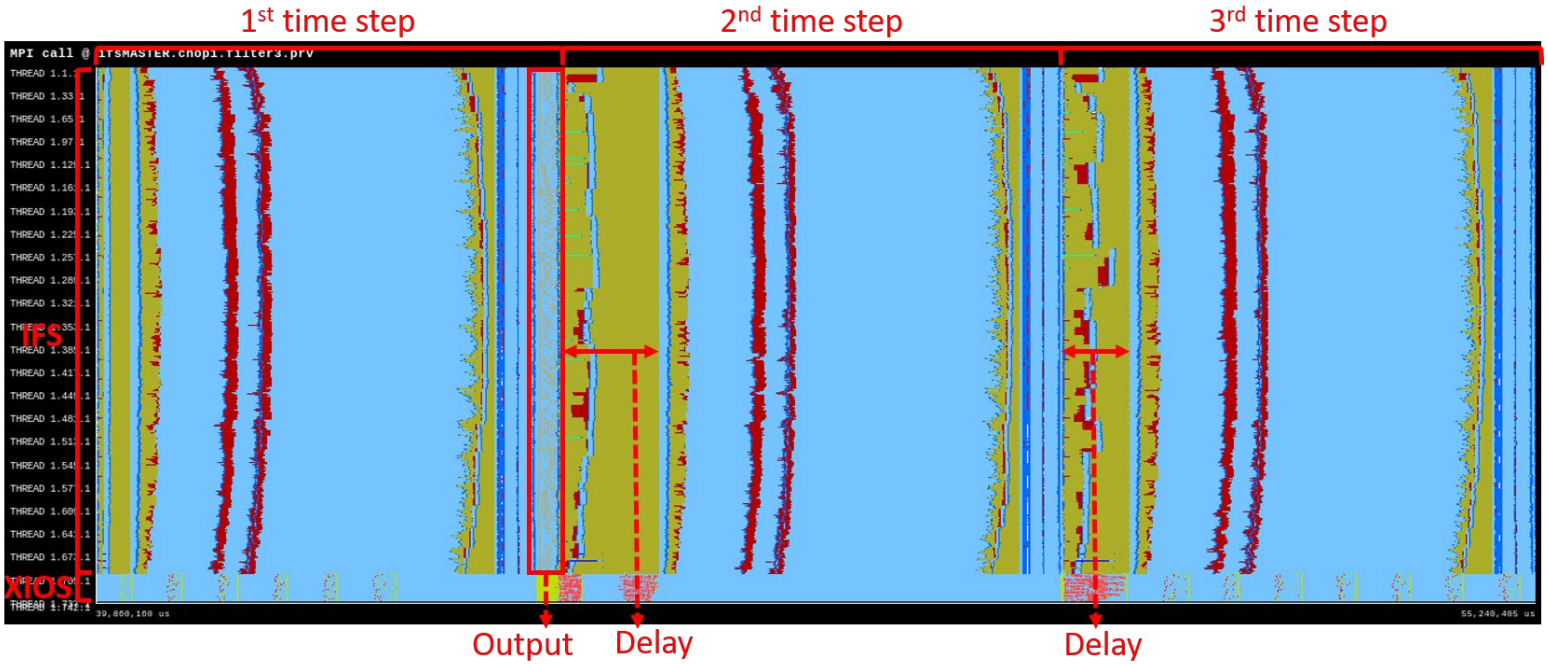
Output

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 12:24:55 | 0AAA00AAA | STEPO | 318 | 27.370 | 27.370 | 4.592 | 167:53 |
| 12:25:02 | 0AAA00AAA | STEPO | 319 | 39.994 | 39.994 | 6.708 | 168:33 |
| 12:25:07 | 0AAA00AAA | STEPO | 320 | 28.826 | 28.826 | 4.826 | 169:02 |
| 12:25:12 | 0AAA00AAA | STEPO | 321 | 28.034 | 28.034 | 4.701 | 169:30 |
| 12:25:16 | 0AAA00AAA | STEPO | 322 | 27.770 | 27.770 | 4.655 | 169:58 |
| 12:25:21 | 0AAA00AAA | STEPO | 323 | 27.690 | 27.690 | 4.654 | 170:26 |
| 12:25:26 | 0AAA00AAA | STEPO | 324 | 27.854 | 27.854 | 4.679 | 170:53 |
| 12:25:33 | 0AAA00AAA | STEPO | 325 | 42.771 | 42.771 | 7.158 | 171:36 |
| 12:25:38 | 0AAA00AAA | STEPO | 326 | 30.114 | 30.114 | 5.044 | 172:06 |
| 12:25:43 | 0AAA00AAA | STEPO | 327 | 30.870 | 30.870 | 5.181 | 172:37 |
| 12:25:48 | 0AAA00AAA | STEPO | 328 | 27.874 | 27.874 | 4.682 | 173:05 |

# Overlapping computation and communication



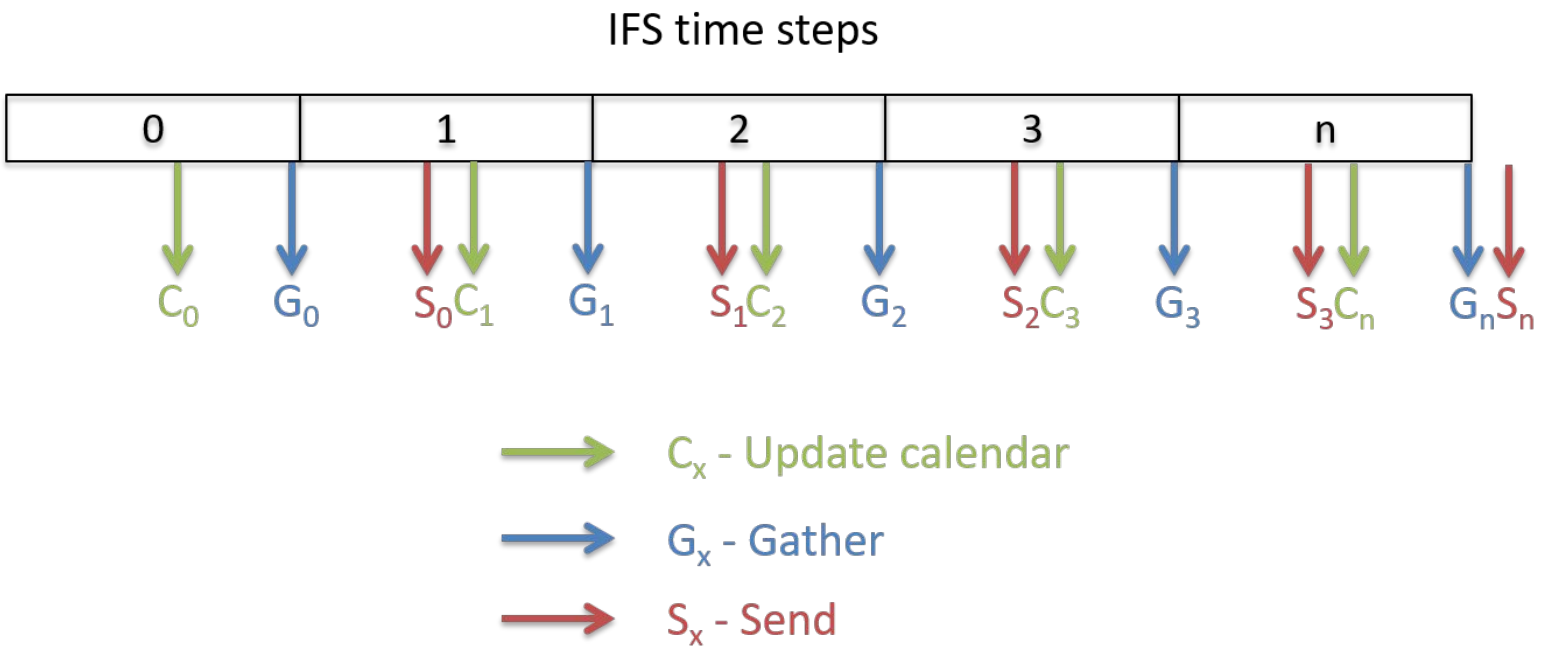
- The trace shows that after an output time step, there is a delay in the communication of the next two time steps (*MPI_Waitany* and *MPI_Alltoallv*)
- There is a conflict between intra IFS communication and IFS to XIOS communication



# 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



- This new scheme improves the execution time of the three time steps that follow an output time step:

Non-output

Output

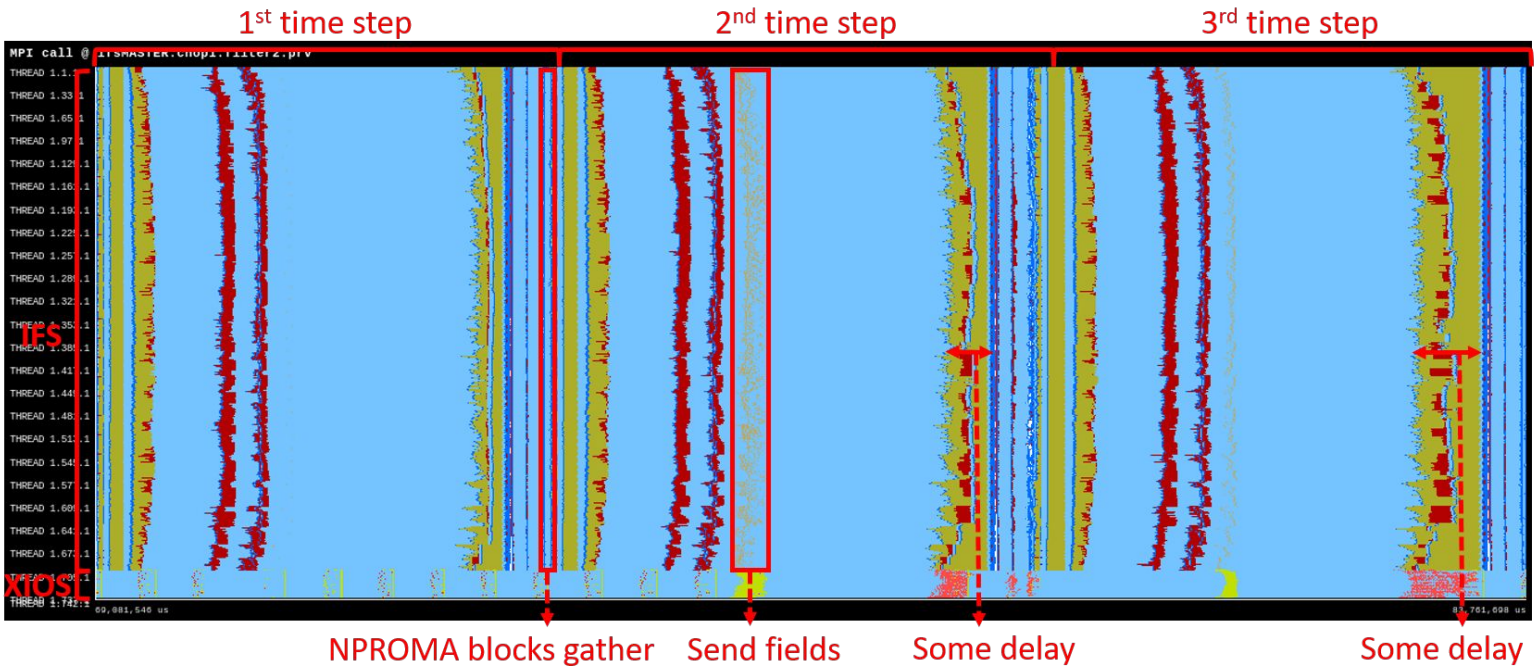
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 12:27:45 | 0AAA00AAA | STEPO | 318 | 26.926 | 26.926 | 4.514 | 162:23 |
| 12:27:52 | 0AAA00AAA | STEPO | 319 | 38.414 | 38.414 | 6.441 | 163:01 |
| 12:27:56 | 0AAA00AAA | STEPO | 320 | 27.054 | 27.054 | 4.535 | 163:28 |
| 12:28:01 | 0AAA00AAA | STEPO | 321 | 27.030 | 27.030 | 4.534 | 163:55 |
| 12:28:05 | 0AAA00AAA | STEPO | 322 | 26.882 | 26.882 | 4.502 | 164:22 |
| 12:28:10 | 0AAA00AAA | STEPO | 323 | 27.394 | 27.394 | 4.607 | 164:50 |
| 12:28:15 | 0AAA00AAA | STEPO | 324 | 27.142 | 27.142 | 4.549 | 165:17 |
| 12:28:21 | 0AAA00AAA | STEPO | 325 | 39.310 | 39.310 | 6.579 | 165:56 |
| 12:28:26 | 0AAA00AAA | STEPO | 326 | 28.318 | 28.318 | 4.755 | 166:24 |
| 12:28:31 | 0AAA00AAA | STEPO | 327 | 28.686 | 28.686 | 4.813 | 166:53 |
| 12:28:35 | 0AAA00AAA | STEPO | 328 | 26.990 | 26.990 | 4.527 | 167:20 |

- The execution time is reduced 122 seconds, from 7629 seconds to 7507 seconds

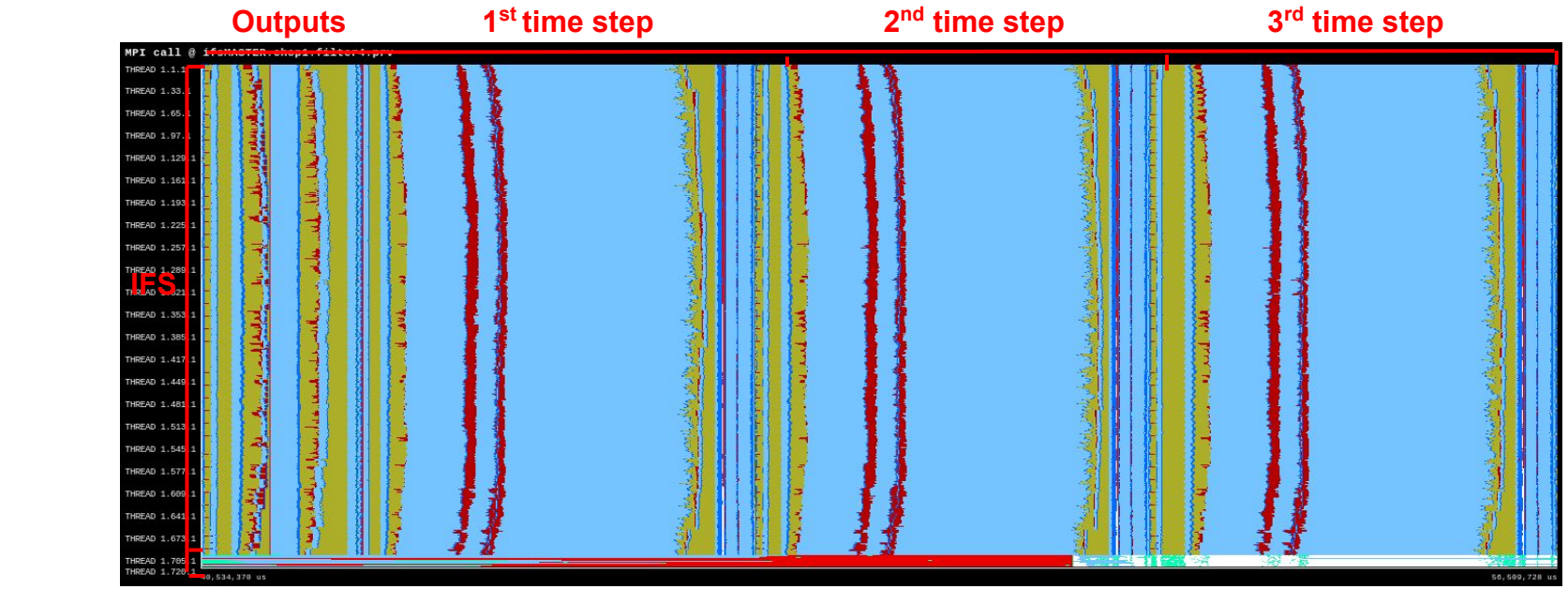
# Overlapping computation and communication



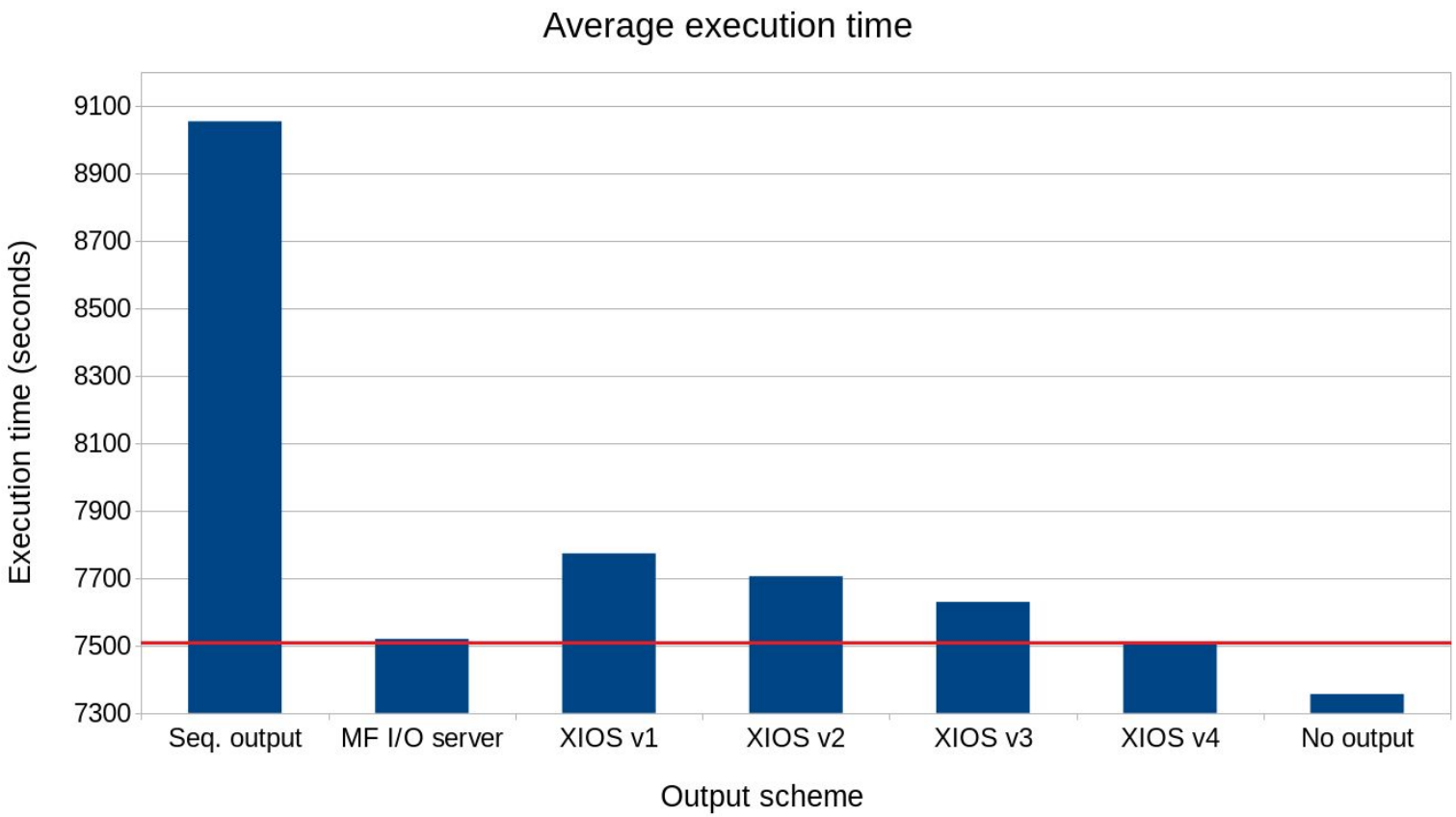
- The trace shows that there is no delay at the beginning of the 2nd and 3rd time steps
- However, there is some delay at the end, but it is less significant



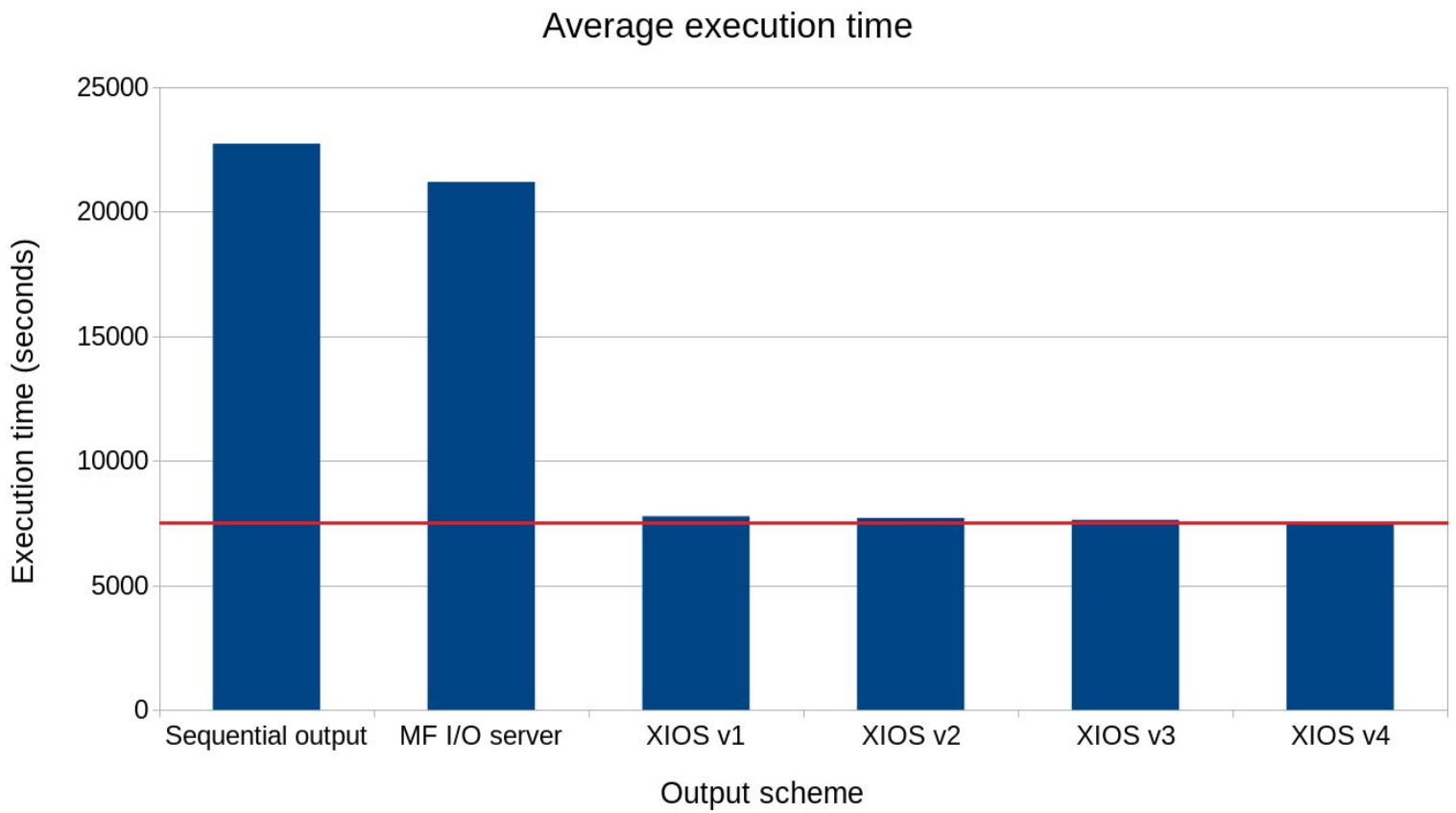
- The trace shows that after an output time step using MF IO server, there is NO delay in the transformation communications or the next two time steps.



# Comparison test



# Comparison test adding GRIB to netCDF post-processing



# Conclusions

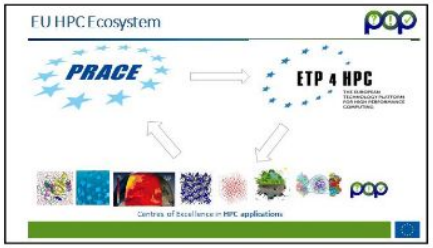


- The integration with no optimization already improved the execution time:
  - Sequential output 9054 seconds (23% of overhead) → IFS-XIOS integration 7773 seconds (5.6% of overhead)
- Performance highlights of the most optimized version:
  - It is slightly faster than the MF I/O server (but without FullPos calls): 7519 s vs. 7507 s
  - It is only 151 seconds slower than no output (2% of overhead)
  - Within 151 seconds IFS outputs 3.2 TB of data
- When post-processing to convert GRIB to netCDF files is taken into account:
  - The post-processing takes 13680 seconds (3.8 hours)
  - Thus, the most optimized version is a 202% faster than the sequential output and a 182% faster than the MF I/O server

www.bsc.es


# Thank you!

# POP CoE




- A **Centre of Excellence**
  - On **Performance Optimisation and Productivity**
  - Promoting **best practices in parallel programming**
- Providing **Services**
  - Precise understanding of application and system behaviour
  - Suggestion/support on how to refactor code in the most productive way
- **Horizontal**
  - Transversal across application areas, platforms, scales
- **For (your?) academic AND industrial codes and users !**