

# Balancing EC-Earth3: Improving the performance of EC-Earth CMIP6 configurations by minimizing the coupling cost

Mario Acosta<sup>1</sup>, Sergi Palomas<sup>1</sup>, and Etienne Tourigny<sup>1</sup>

<sup>1</sup>Barcelona Supercomputing Center

February 27, 2023

## Abstract

Earth System Models (ESMs) are complex systems used in weather and climate studies generally built from different independent components responsible for simulating a specific realm (ocean, atmosphere, biosphere, etc.). To replicate the interactions between these processes, ESMs typically use coupling libraries that manage the synchronization and field exchanges between the individual components, which run in parallel as a Multi-Program, Multiple-Data (MPMD) application.

As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in HPC platforms has become increasingly challenging and of major concern. One of the critical bottlenecks is the load-imbalance, where the fastest components will have to wait for the slower ones. Finding the optimal number of processing elements (PEs) to assign to each of the multiple independent constituents to minimize the performance loss due to synchronizations and maximize the overall parallel efficiency is impossible without the right performance metrics, methodology and tools.

This paper presents the results of balancing multiple Coupled Model Intercomparison Project phase 6 (CMIP6) configurations for the EC-Earth3 ESM. We will show that intuitive approaches can lead to suboptimal resource allocations and propose new setups up to 25% faster while reducing the computational cost by 72%.

We prove that new methods are needed to deal with the load-balance of ESMs and hope that our study will serve as a guide to optimize any other coupled system.

1

# Balancing EC-Earth3

2

## Improving the performance of EC-Earth CMIP6 configurations by minimizing the coupling cost

3

4

M. C. Acosta

5

S. Palomas

6

E. Tourigny

7

Barcelona Supercomputing Center

8

### Key Points:

9

- Find what are the most useful performance metrics and how to understand them in order to tackle the load-imbalance problem in coupled ESMs

10

11

- Understand the individual scalability properties of the components and their relationships in multiple EC-Earth3 CMIP6 configurations

12

13

- Contrast our approach against simpler and traditional ones that the community use to deal with the load-balance problem

14

---

Corresponding author: M. C. Acosta, [mario.acosta@bsc.es](mailto:mario.acosta@bsc.es)

Corresponding author: S. Palomas, [sergi.palomas@bsc.es](mailto:sergi.palomas@bsc.es)

Corresponding author: E. Tourigny, [etienne.tourigny@bsc.es](mailto:etienne.tourigny@bsc.es)

**Abstract**

Earth System Models (ESMs) are complex systems used in weather and climate studies generally built from different independent components responsible for simulating a specific realm (ocean, atmosphere, biosphere, etc.). To replicate the interactions between these processes, ESMs typically use coupling libraries that manage the synchronization and field exchanges between the individual components, which run in parallel as a Multi-Program, Multiple-Data (MPMD) application. As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in HPC platforms has become increasingly challenging and of major concern. One of the critical bottlenecks is the load-imbalance, where the fastest components will have to wait for the slower ones. Finding the optimal number of processing elements (PEs) to assign to each of the multiple independent constituents to minimize the performance loss due to synchronizations and maximize the overall parallel efficiency is impossible without the right performance metrics, methodology and tools. This paper presents the results of balancing multiple Coupled Model Intercomparison Project phase 6 (CMIP6) configurations for the EC-Earth3 ESM. We will show that intuitive approaches can lead to suboptimal resource allocations and propose new setups up to 25% faster while reducing the computational cost by 72%. We prove that new methods are needed to deal with the load-balance of ESMs and hope that our study will serve as a guide to optimize any other coupled system.

**Plain Language Summary**

Earth System Models (ESMs) are complex systems used in weather and climate studies generally built from different independent components responsible for simulating a specific realm (ocean, atmosphere, biosphere, etc.). To replicate the interactions between these processes, ESMs communicate during the simulation to exchange data. As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in HPC platforms has become increasingly challenging and of major concern. One of the critical bottlenecks is the load-imbalance, where the fastest components will have to wait for the slower ones. Finding the optimal number of processing elements (PEs) to assign to each of the independent components without the right performance metrics, methodology and tools. We will show that intuitive approaches can lead to suboptimal setups and propose new ones up to 25% faster while reducing the computational cost by 72%. Thus, proving that new methods are needed to deal with the load-balance of ESMs and hope that our study will serve as a guide to optimize any other coupled system.

## 1 Introduction

The load-balance of Earth System Models (ESMs), where different components (ocean, atmosphere, land, sea ice, etc.) are running concurrently, is increasingly complex as we keep introducing more features during the simulation. Although some models run the different components in sequential mode, the most common case in the community is the execution of the different components in parallel (separate cores) and using a coupler to synchronize the components and exchange information among them. These kinds of applications are computationally known as MPMD (Multiple Program, Multiple Data), where different binaries are executed in parallel simulating a natural phenomena using a parallel paradigm such as MPI. Given the nature of the physics underneath, the components within the system have to interact during the simulation (i.e. coupled). Running coupled ESM adds significant changes in the computational performance:

- Coupling data must be interpolated (regridded), adding extra computation compared to standalone runs. Furthermore, interpolation constraints such as conservation could require serialization techniques and may reduce the parallel efficiency of the model
- Coordination among components is needed to exchange data between them. Depending on the setup, faster components will have to wait for the slower ones, resulting in IDLE processes and an overall reduction of the parallel efficiency
- The speed (i.e. parallelization) at which each independent has to run to minimize the cost of the synchronizations will depend on the coupled configuration. Components can no longer run at their optimal scalability point but rather at the optimal point for the whole system

As we will show, the waiting time due to the synchronization between multiple components in a coupled ESM can have a negative effect on the performance achieved. We have observed that it can consume up to 75% of the total computational cost of the simulation. In this work, we present the methodology used and the results obtained to balance different Coupled Model Intercomparison Project phase 6 (CMIP6) configurations of EC-Earth3, each one having its own particularities (irregular timesteps, memory requirements, different scalability properties, processor mapping, etc.). The solutions achieved to obtain the best possible Processing Elements (PEs) setup will be contrasted with traditional methods that are often adopted but, as we will see, can lead to a waste of computational resources. Furthermore, as we show how to optimize setups under distinct contexts by exploring multiple configurations of EC-Earth3, we hope that the results will help with new load-balancing studies of other ESMs, as we have examined common patterns of this type of applications. To achieve our goals, we needed to extend the current set of metrics in the Computational Performance for Model Intercomparison Projects (Balaji et al., 2017) (CPMIP) and tools to get them.

## 2 Related work

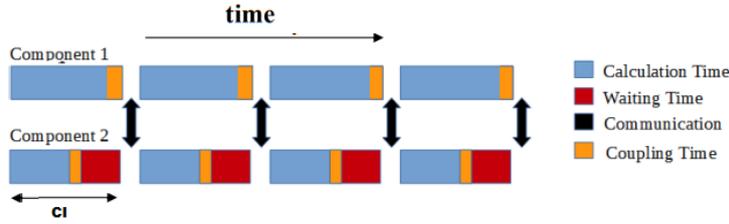
Models that simulate the Earth's climate are among the most computationally-intensive applications that run on HPC platforms nowadays. Still, the performance that these models achieve is far from ideal as shown by Balaji (2015), and one of the main limiting factors is the load-imbalance. Valcke et al. (2012) have shown that many of the current climate applications used in several institutions are built from different individual components and their interactions are managed by a coupler. Although different approaches exist to couple the components in an ESM, one of the most commonly used is to keep each component as a separate binary and use the coupling library API calls to transform and exchange the fields during the simulation. The coupling library ensures the synchronization and regridding processes. Some notable examples are the OASIS3-MCT (Valcke, 2013), C-Coupler (Liu et al., 2014) and YAC (Hanke et al., 2016) couplers. EC-Earth3

100 (Döscher et al., 2022) is a very well-known ESM used in many European institutions and  
 101 uses the OASIS3-MCT coupling library. During the simulation, different components ex-  
 102 change fields and they must be synchronized, rising the load-imbalance problem. The  
 103 fastest components will have to wait for the slower ones to finish before sending/receiving  
 104 the data. The process of finding the best number of PEs to assign to each one of the com-  
 105 ponents which minimizes the overall performance loss due to the synchronization among  
 106 multiple binaries is known as balancing a coupled ESM. An example of dealing with the  
 107 load-imbalance has been shown by Will et al. (2017) for the COSMO-CLM regional cli-  
 108 mate model. Like in EC-Earth3, this ESM uses OASIS3-MCT to couple the multiple bi-  
 109 naries (atmosphere, ocean, etc.) and they used the LUCIA tool (Maisonave et al., 2020)  
 110 to find the optimal number of processes for each component considering the simulation  
 111 time, energy cost and parallel efficiency of the simulation. They stated that the coupling  
 112 time will be minimum if one finds an allocation in which all components run at the same  
 113 speed. The approach consisted of 1) finding a setup in which all components run at the  
 114 same speed with few resources, 2) doubling the number of PEs assigned to each com-  
 115 ponent, 3) readjusting the PEs given to each component so that they run at the same  
 116 speed again, 4) loop to 2 if none of the components' parallel efficiency is below 50%. Even  
 117 though this approach is simple, intuitive and the most frequently used by the commu-  
 118 nity, we will show that it can lead to suboptimal setups. Donners et al. (2012) showed  
 119 that the coupling overhead was an important limiting factor of EC-Earth3 performance.  
 120 They found out that the results obtained from the LUCIA tool could be misleading and  
 121 the approach of running the ocean and atmospheric components at the same speed was  
 122 not good enough to reduce the coupling cost to the minimum. Moreover, they also stud-  
 123 ied in (Acosta et al., 2016) the computing cost of using conservative remapping algorithms  
 124 in the coupler.

125 To analyse the performance of ESMs and evaluate the overhead due to the coupling,  
 126 we will need the right set of performance metrics. As noted by Balaji et al. (2017), given  
 127 the heterogeneity of HPC platforms on which these models run, the differences between  
 128 multiple implementations of ESMs and the varying configurations that can be used, typ-  
 129 ical performance metrics like the FLOPS, cache miss ratio, etc. may not be sufficient for  
 130 the whole range of ESMs. This led to the proposal of the Computational Performance  
 131 Model Intercomparison Project (CPMIP) metrics, which are a collection of metrics es-  
 132 pecially designed for ESMs. In this article, we will use and extend some of them to ac-  
 133 curately address the load-imbalance problem.

### 134 **3 Coupled ESMs**

135 One of the most used approaches to couple ESMs is to keep each individual com-  
 136 ponent as an independent code and use a coupling library (such as OASIS) that deals  
 137 with all the communication between them. This coupling approach is referred to as us-  
 138 ing an "external coupler or coupling library". While it offers the advantage that the changes  
 139 in the source code of each component needed to build the coupled ESM are minimum  
 140 (i.e. keeping independently developed codes self-contained) this implementation has some  
 141 drawbacks to the performance achieved: 1) components will run concurrently on sepa-  
 142 rate PEs and will have to send the exchanged fields across different nodes through the  
 143 HPC network, 2) dependencies between components will reduce the parallel efficiency  
 144 of the ESM as the fastest ones will have to wait for the slowest, 3) an extra computa-  
 145 tion may be needed to transform the data from one component grid to another before  
 146 sending the coupled field. Figure 1 shows the common coupling pattern between two com-  
 147 ponents using an external library. Reducing the IDLE time due to the synchronization  
 148 between components is of utmost importance to achieve a well-balanced ESM and use  
 149 the HPC resources effectively.



**Figure 1.** Overview of two independent components using a coupling library to build an ESM. Each component runs on separate PEs. At the end of each coupling interval (CI), both components need to exchange some coupling fields. The calculation time of Component 2 is less (in blue) and has to wait (in red) for Component 1, which is slower. Furthermore, the execution of both components is extended due to interpolation (in orange) and some fields are exchanged before starting the next CI (black arrows)

150

### 3.1 EC-Earth3 coupling configurations

151

152

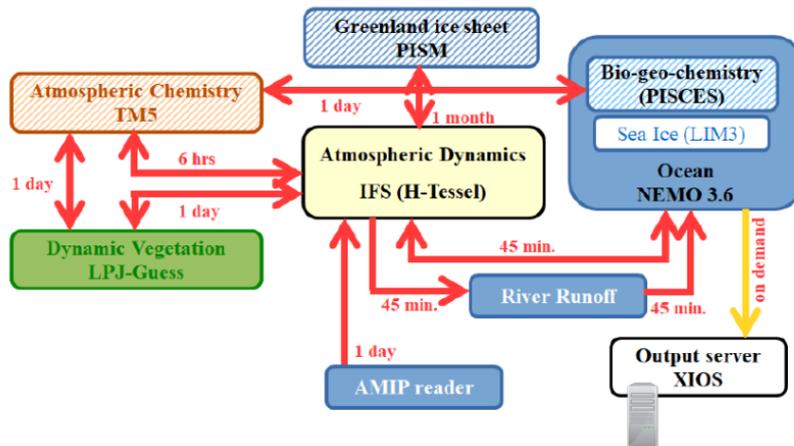
153

154

155

156

The ESM for which we have conducted the load-balance studies is EC-Earth3 (Döscher et al., 2022) which was used in the Coupled Model Intercomparison Project phase 6 (CMIP6) project. The EC-Earth Consortium brings together 27 research institutes from 10 European countries to collaborate on the development of the EC-Earth3 ESM. EC-Earth3 is a fully coupled Atmosphere-Ocean-Land-Biosphere model, that can be used in seasonal to decadal predictions and climate change projections.



**Figure 2.** Overview of EC-Earth3 with the coupling links between all components that can be coupled (Döscher et al., 2022).

157

158

159

160

161

162

As shown in Figure 2, there are multiple possible configurations in EC-Earth3 depending on the individual components used, which are: IFS for the atmosphere and land surface; NEMO for the ocean, sea ice (LIM3 module) and biogeochemistry (PISCES module); LPJ-GUESS (hereafter named LPJG) for the dynamic vegetation; and TM5 for the Atmospheric composition. Each component can run in standalone mode and uses different grids and input data. During this work, we have studied the following 4 different

163 EC-Earth3 coupled configurations where these components run in parallel using separate processors:  
164

- 165 • EC-Earth3: IFS + NEMO
- 166 • EC-Earth3-Veg: IFS + NEMO + LPJG
- 167 • EC-Earth3-AerChem: IFS + NEMO + TM5
- 168 • EC-Earth3-CC: IFS + NEMO + LPJG + TM5\_CO2 + PISCES

169 The resolution used for these configurations is TL255-ORCA1, corresponding to  $\tilde{80}$ km  
170 for the atmosphere,  $1^\circ$  for the ocean, with a coupling and component timestep frequency  
171 of 2700s (45min) for IFS and NEMO. All the coupling process is handled by the OASIS3-  
172 MCT coupling library. In addition, NEMO uses the XIOS library to allow having multiple  
173 IO servers that manage the output independently from the model processors and  
174 there is the RiverRunoff process that collects surface and sub-surface runoff from IFS  
175 and eventually sends this as runoff to NEMO. None of them are significant for the load-  
176 balance and we chose to include them in our analysis. For more information about these  
177 configurations, please refer to Döscher et al. (2022).

### 178 3.2 Environment

179 All simulations have been executed in the Barcelona Supercomputing Center HPC  
180 machine MareNostrum4, using Intel Xeon Platinum 8160 processors from the Skylake  
181 generation. It is a Lenovo system composed of SD530 Compute Racks, an Intel Omni-  
182 Path high-performance network interconnect and running SuSE Linux Enterprise Server  
183 as the operating system. Its current Linpack Rmax Performance is 6.23 Petaflops. This  
184 general-purpose block consists of 48 racks housing 3456 nodes with 48 PEs each. Giving  
185 a grand total of  $3456 * 48 = 165,888$  processor cores and 390 Terabytes of main  
186 memory.

### 187 3.3 Performance metrics

188 CPMIP (Balaji et al., 2017) are a collection of performance metrics used to evaluate  
189 the performance of ESMs. The ones used in this work are:

- 190 • Runtime (T): The total execution time of the run
- 191 • Parallelization (P): The number of PEs allocated for the run
- 192 • SYPD: The number of Simulated Years per Day (24 hours of executing time on  
193 the HPC platform)
- 194 • CHSY: The number of Core-Hours per Simulated Year
- 195 • Coupling cost: The fractional cost associated with the coupling events. This in-  
196 cludes the time waiting, sending and interpolating the data.

$$197 \quad Cpl\_cost = \frac{TP - \sum_c T_C P_C}{TP} \quad (1)$$

197 Where  $T_C$  and  $P_C$  are the runtime and parallelization of each component.

198 Additionally, we have introduced the component coupling cost (Component\_cpl\_cost),  
199 which measures how much each component adds to the overall  $Cpl\_cost$ .

$$200 \quad Component\_cpl\_cost = \frac{T_{C_{cpl}} P_C}{TP} \quad (2)$$

200 The OASIS3-MCT coupling library will record the starting and ending times of each cou-  
201 pling event (waiting, sending, interpolating). After the run, this timing information is  
202 post-processed using the LUCIA (Maisonave et al., 2020) tool to collect the mentioned  
203 CPMIP metrics for the simulation.

## 204 4 Results

205 In this section, we show the results obtained when balancing multiple EC-Earth3  
 206 configurations currently used for various climate studies. Firstly, we will discuss two dif-  
 207 ferent solutions for the most common experiment, consisting of IFS coupled with NEMO  
 208 (EC-Earth3 SR). Then we will analyze how introducing the TM5 component in the EC-  
 209 Earth3-AerChem configuration limits the coupled model scaling. Finally, we show which  
 210 is the best approach to allocate the LPJG processes in Carbon-cycle experiments, includ-  
 211 ing the modifications in the total number of processes in EC-Earth3-Veg and EC-Earth3-  
 212 CC configurations. For the results, we have used a high-priority queue. Although it re-  
 213 duces queuing time by having near-instant access to the HPC resources, it imposes two  
 214 constraints: The maximum number of concurrent resources for a job and user is limited  
 215 to 768 PEs ( $768/48 = 16$  nodes) and the wall-clock time is limited to 2h. We include  
 216 this information as relevant to prove that restrictions such as parallel resources or job  
 217 max duration can be managed with the proper methodology.

### 218 4.1 EC-Earth3 SR: IFS-NEMO

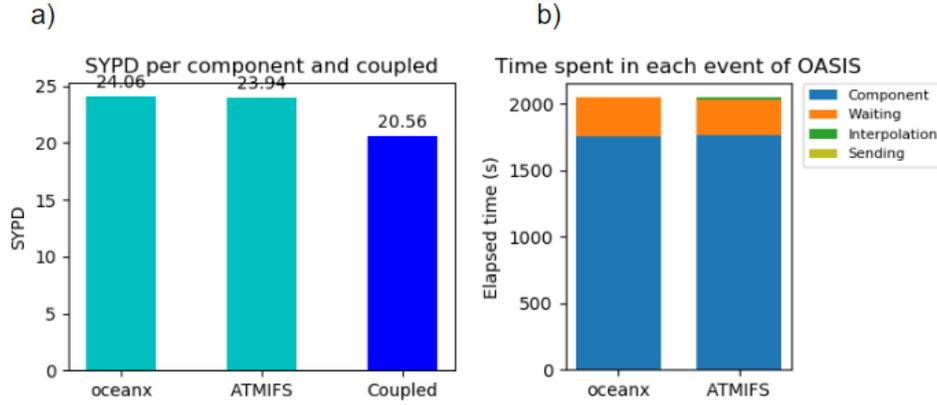
219 In this part, we will compare the typical approach used for the community (same  
 220 SYPD) versus the new one proposed in this work.

#### 221 4.1.1 Same SYPD approach

222 A common approach consists of finding a configuration in which all components  
 223 run at the same speed (i.e. SYPD). If we can achieve this, the waiting time due to model  
 224 synchronizations would ideally be 0. Figure 3 shows a setup for which IFS and NEMO  
 225 run at the same SYPD by using 11 ( $48 \cdot 11 = 528$  processes) and 4 nodes ( $48 \cdot 4 = 192$   
 226 processes) for IFS and NEMO, respectively. This is the fastest possible configuration where  
 227 both components' SYPD are similar and fit into the 16-node high-priority queue limita-  
 228 tion (15 for both components and 1 reserved for XIOS). As we see, however, the cou-  
 229 pled SYPD is much lower than the one expected, which theoretically should be approx-  
 230 imately as fast as the lowest SYPD achieved by each independent component.

231 Figure 3 b) shows the time spent for the component execution, waiting, interpo-  
 232 lation and sending for IFS and NEMO for this configuration, obtained through the LU-  
 233 CIA tool. It shows that both components take approximately the same time to finish  
 234 their own execution (in blue) as expected, given that both run at the same SYPD. But  
 235 surprisingly, both components' coupled execution is extended due to the time lost wait-  
 236 ing. Since those are the only two coupled components in this configuration, this can only  
 237 mean that they are waiting for each other.

238 We can see the detailed coupled information per timestep in Figure 4. NEMO (at  
 239 the top) time-stepping (in blue) is regular during the simulation. Meanwhile, every 3 hours  
 240 of simulation (4 timesteps) IFS (at the bottom) component timestep takes much longer  
 241 than the others. Having a component with irregular timestep lengths is quite common.  
 242 Here we know that is due to IFS computing the radiation but can also happen due to  
 243 IO operations or ice calculation. As we forced both components to run at the same SYPD,  
 244 we have created a cyclical conflict in which every 4 timesteps NEMO will have to wait  
 245 for IFS, while IFS is waiting for NEMO during the other timesteps. Moreover, the wait-  
 246 ing time in the 4th timestep in NEMO equals the sum of the waiting time of the pre-  
 247 vious 3 timesteps in IFS. This pattern is repeated during the whole execution making  
 248 the coupled solution slower than expected (i.e. as fast as the slowest component).



**Figure 3.** Results using the same SYPD strategy in EC-Earth3-SR experiments. a) NEMO (oceanx) and IFS (ATMIFS) components running at the same SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component



**Figure 4.** Timestep component and coupling (Waiting, Interpolation, Sending) times for IFS and NEMO when running at the same SYPD

249

#### 4.1.2 Optimal solution

250

251

252

253

254

255

256

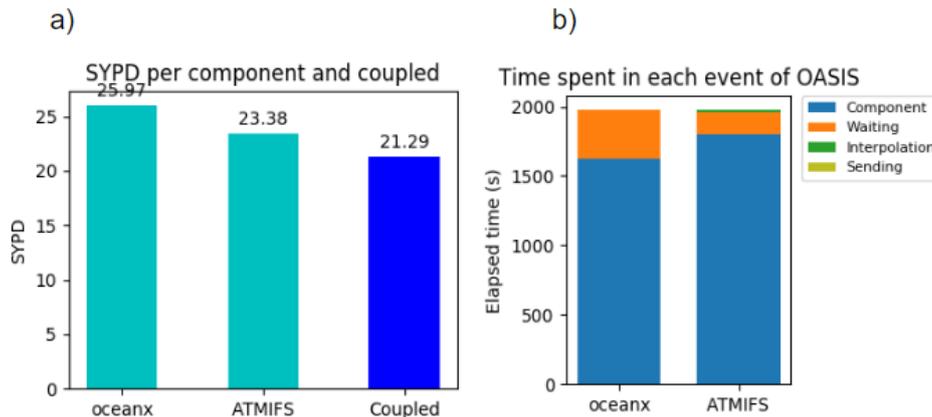
257

As we have seen above, the performance of EC-Earth SR experiments when running IFS and NEMO at the same SYPD seems to be suboptimal since the coupled SYPD is noticeably lower than that of its constituents. As a consequence of forcing both components to have the same computational time, the waiting time due to the coupling synchronizations is also equal (see Figure 3), and the resulting coupling cost is 13.3%. Even though the IFS timestep irregularities observed in Figure 4 imply that it is impossible to find a configuration which reduces the coupling cost to 0, it is still possible to reduce the coupling cost and use the resources more effectively.

**Table 1.** Performance results for EC-Earth-SR experiments when using the same SYPD and our approach

	same SYPD	optimum
SYPD	20.5	21.3
CHSY	896	863
Cpl cost (%)	13.3	10.8
PEs	720	720

258 After running the LUCIA tool, we can take a look at the *Component\_cpl\_cost* met-  
 259 ric (Equation 2) and see that 3.6% of the total coupling cost is due to NEMO waiting  
 260 for IFS and 9.7% due to IFS waiting for NEMO ( $9.7 + 3.6 = 13.3$ ). This means that the  
 261 computing cost of the waiting time is much higher in IFS than in NEMO. Something that  
 262 we were expecting since IFS is using more PEs than NEMO and, even though the wait-  
 263 ing time is the same in both components, more processors are IDLE when IFS has to  
 264 wait. Therefore, it is preferable to give some resources from IFS to NEMO so that IFS  
 265 will wait for less time (NEMO will run faster) and fewer processes will be IDLE. Or in  
 266 other words, we want to reduce the *Component\_coupling\_cost* of the component with  
 267 the highest value for this metric. Moreover, from our single-component scalability anal-  
 268 ysis, we know that NEMO scales better in this range of processor counts. Following this  
 269 approach, we find a new setup with 504 PEs for IFS and 216 for NEMO (we have taken  
 270 24 PEs away from IFS and are now used by NEMO). Thus, the total number of resources  
 271 remains the same but, as we see in Figure 5 a), NEMO is now a bit faster than IFS. With  
 272 this setup, the *Component\_cpl\_cost* of both components is almost the same. This means  
 273 that even though NEMO waiting time is more than twice that of IFS (see Figure 5 b)),  
 274 the cost is the same as it uses fewer PEs. The results achieved are summarized in Ta-  
 275 ble 1. With our analysis, we have found a setup which is 4% faster ( $21.3/20.5$ ) without  
 276 adding any extra resources but only by properly reallocating the PEs from one compo-  
 277 nent to the other using the *Component\_coupling\_cost* metric. As a consequence, the us-  
 278 age of the resources is also better (the CHSY and the coupling cost have also been re-  
 279 duced significantly).

**Figure 5.** Results using an optimal EC-Earth-SR resource configuration. a) NEMO, IFS and Coupled SYPD using 504 and 216 PEs respectively. b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO and IFS using 216 and 504 PEs respectively

280

## 4.2 EC-Earth3-AerChem

281

282

283

284

285

286

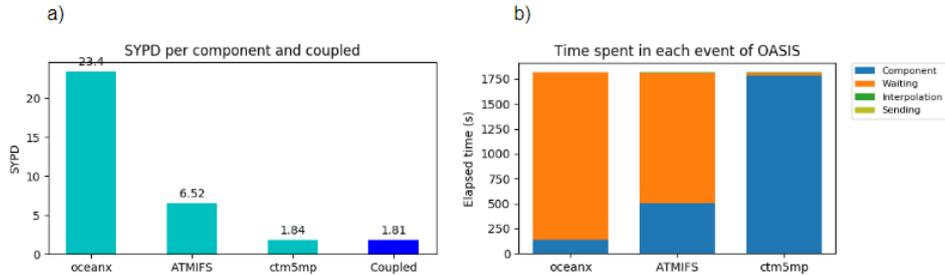
287

288

289

290

The addition of TM5 to EC-Earth in the EC-Earth3-AerChem configuration caused a drastic decrease in the total SYPD that was achieved decreased dramatically. This is mainly because TM5 is very slow compared to the other models and does not scale, being the dominant bottleneck for the coupled simulation as the components have to be synchronized (through the exchange of some particular fields) at each coupling timestep. Moreover, TM5 limits the maximum number of processes that IFS can use to 256 due to the way that spectral fields are exchanged. The default setup previous to this study was using 45 processes for TM5, 256 for IFS and 240 for NEMO. Figure 6 b) shows the magnitude of the overhead introduced by the TM5 component. We also see in Figure 6 a) that this is mainly happening due to IFS and NEMO being much faster than TM5.



**Figure 6.** Results using the default EC-Earth3-AerChem resource configuration. a) NEMO, IFS, TM5 and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO, IFS and TM5 (ctm5mp)

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

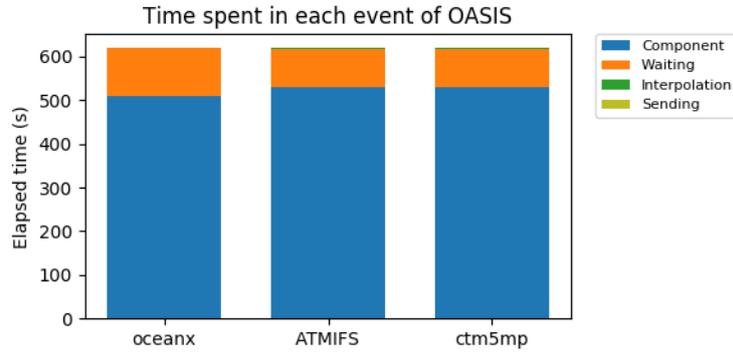
314

315

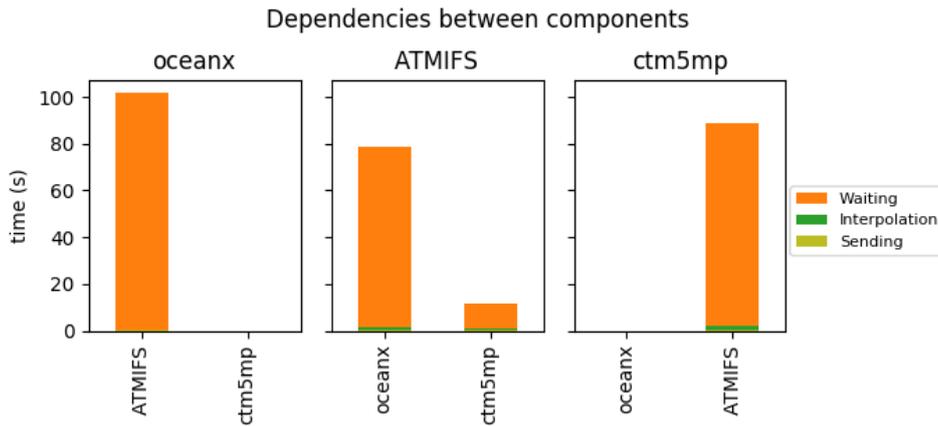
316

The first course of action to optimize the setup for the AerChem configuration was to make TM5 faster, but the scalability tests revealed that this component can barely scale. Nonetheless, we found that it was better to use 90 processes instead of the 45 that were originally allocated, obtaining a speedup of 1.35x. Since we could not further increase the SYPD for this configuration, we decided to save as many cores (and energy) as possible by reducing the number of resources given to the other components, which in this case are IFS and NEMO. Following the same SYPD approach, we end up with a setup that uses 80 PEs for IFS, 16 for NEMO and 90 for TM5, giving a total SYPD of 1.97 and 620 CHSY with a coupling cost of 14.7%. While this configuration is much better than the default one, we did not stop here but rather tried to push the setup a little bit further using our approach with the `Component_cpl_cost`. As we see in Figure 7, we again see that all components' execution time is the same (in blue) but at the same time, they are all waiting during a noticeable amount of time (in orange). Thus, as we saw in Section 4.1, the resulting configuration is less efficient than it could potentially be. By looking at Figure 8 we see that running at the same speed is not only bad for IFS and NEMO, but also for TM5 given that IFS and TM5 are waiting for each other as well. Using this information and the `Component_cpl_cost` metric led us to a configuration with 84 processes for IFS (minimizing the waiting time on TM5 due to IFS), 24 for NEMO (minimizing the waiting time on IFS due to NEMO), and 90 for TM5 (the maximum speed for the slowest component). As explained in Section 4.1.2, we kept NEMO faster than IFS to achieve the best possible combination between these two components while staying just above the SYPD achieved by TM5 (see Figure 9), given that this component is the slowest of the three and we prioritize the speed. The figure also shows that with this new setup, IFS and TM5 spent most of their time without having to wait, while NEMO still suffers from the synchronization but this now only has an effect on 24 PEs, achieving a coupled SYPD of 2.26, 2018 CHSY and 8.25% of coupling cost. Therefore, when

317 comparing with the same SYPD strategy, we managed to make the simulation 1.15x faster  
 318 while reducing the CHSY by 7% and cutting the coupling cost by half.



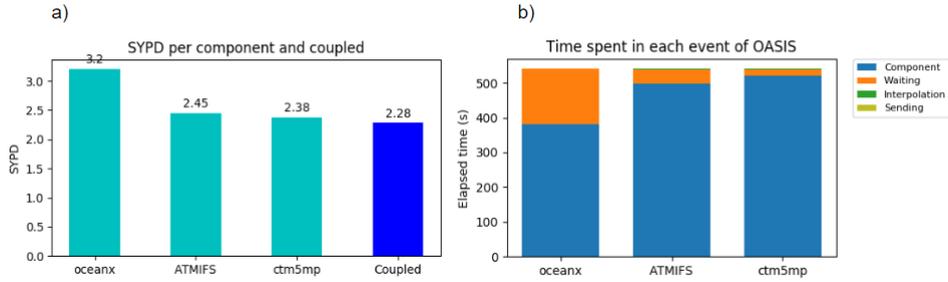
**Figure 7.** Component and coupling (Waiting, Interpolation, Sending) times for an EC-Earth3-AerChem experiment when all components run at the same speed



**Figure 8.** Waiting time between components for an EC-Earth3-AerChem experiment when all components run at the same SYPD. NEMO is always waiting for IFS, IFS is mostly waiting for NEMO and TM5 only waits for IFS

**Table 2.** Performance results of the original, same SYPD and optimal resource setups for EC-Earth-AerChem experiments

	Original	same SYPD	Optimum
SYPD	1.81	1.97	2.26
CHSY	7173	2266	2102
Cpl cost (%)	75.3	14.7	8.25
PEs	541	186	198



**Figure 9.** a) Results using a balanced EC-Earth3-AerChem resource configuration. a) IFS, NEMO, TM5 and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component

319

### 4.3 EC-Earth3-Veg

320

321

322

323

324

325

326

327

This experiment configuration adds LPJG to be coupled with IFS and NEMO. One of the particularities of this component is that it is much faster than those two and, therefore, the strategy of running all components at the same speed can no longer be applied. Figure 10 shows that with the default resource configuration, LPJG spends most of the time waiting, and in Figure 10 we see that this happens because this component is much faster than IFS and NEMO. Ideally, we would like to reduce the number of resources used by LPJG. Still, we found a couple of limitations with this component which have to be taken into account to design an optimal setup for EC-Earth3-Veg configurations:

328

329

330

331

332

333

334

335

336

337

338

339

340

341

- *Memory consumption:* The memory consumption of LPJG is high. On Marens-trum4, it is recommended to use 3 nodes with 96GB of main memory each to ensure that this component won't fail during the simulation due to a lack of memory.
- *Initialization:* Studying the scalability of LPJG we have realized that it is much faster than IFS and NEMO during the execution but it has a slow initialization. We don't need many cores to run LPJG without it interfering with the execution of the other components. However, reducing too much the PEs assigned for LPJG will make the initialization phase slower. This can make hundreds or even thousands of processes (the ones assigned to the other components) wait for the initialization of LPJG at the beginning of the simulation, which can take up to 7 minutes with very few processes. Even though the initialization overhead is mitigated in long simulations, the waste of resources still exists and it could be significant for shorter chunks.

342

343

344

345

346

The only way to reduce the number of PEs used by LPJG while ensuring that it will have access to enough memory is to spread its processes across multiple nodes. Therefore, the use of explicit affinity (to distribute parallel resources through the machine manually) is key to improving this configuration's performance by making it possible to use the memory of multiple nodes, without having to assign all their cores exclusively to LPJG.

347

348

349

To choose whether it is better to share LPJG processes with NEMO or IFS, we have conducted some memory consumption and communication overhead studies for each of these components independently:

350

351

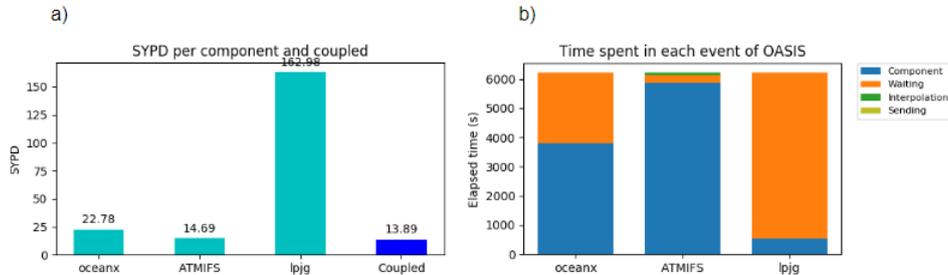
- *The memory consumption:* If IFS or NEMO are consuming too much memory already, LPJG should not share the node with that component. We have tested how

352 the memory consumption of these components changes as we reduce the number  
 353 of cores they use per node.  
 354 • The communication overhead: If the component shows an overhead due to the ex-  
 355 tra communication needed between different nodes after scattering its processes,  
 356 we have to ensure that this loss in efficiency would not be big enough to make the  
 357 explicit affinity solution unworthy.

358 In both cases, IFS and NEMO do benefit from reducing the number of cores per  
 359 node they use and the MPI communication overhead is negligible. According to the re-  
 360 sults obtained, the memory consumption of IFS is higher and we concluded that it is bet-  
 361 ter to make NEMO and LPJG components share resources. The optimal setup uses 336  
 362 processes for IFS, 380 for NEMO and 40 for LPJG. A total of 8 nodes are used by NEMO  
 363 and LPJG at the same time, the first running on 43 cores and the latter on the remain-  
 364 ing 5. Not only is this configuration more efficient, but it also now fits into the debug  
 365 queue as it uses less than 16 nodes and the time to solution for 1-year simulations is less  
 366 than 2h. The results are shown in Table 3. Although the coupled SYPD achieved is 5%  
 367 (13.2/13.9) lower, the number of resources needed has decreased by 40% (768/1104) and  
 368 the CHSY is now 28% (1318/1824) better.

**Table 3.** Performance results of the original and optimal resource setup for EC-Earth-Veg experiments

	Original	Optimum
SYPD	13.9	13.2
CHSY	1824	1318
Cpl cost (%)	28.7	21
PEs	1104	768



**Figure 10.** a) Results using the default EC-Earth3-Veg resource configuration. a) NEMO, IFS, LPJG and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component

#### 369 4.4 EC-Earth3-CC

370 The last of the configurations to evaluate consists again of IFS, NEMO and LPJG  
 371 but it also adds a reduced version of TM5 to simulate the atmospheric Carbon cycle (TM5\_CO2).  
 372 Again, with LPJG we can not use the same SYPD strategy. Instead, we will again show  
 373 how spreading the physical allocation of its processes is the best approach to minimize  
 374 the performance loss by this component and how to balance an experiment with 4 cou-  
 375 pled components. TM5\_CO2 does not scale very well, but after doing the scalability anal-

**Table 4.** Performance results of the original and optimal resource setup for EC-Earth-CC experiments

	Original	Optimum
SYPD	7.1	7.73
CHSY	2104	1428
Cpl cost (%)	33.7	15.1
PEs	621	476

376 ysis we found that it is much faster than the full TM5 execution and that instead of us-  
377 ing 45 processes (as the default resource configuration suggested), this component is faster  
378 when using only 8 processes, achieving almost 9 SYPD. Given that TM5 limits the ex-  
379 ecution speed, we have reduced IFS and NEMO processes to 256 and 192 respectively,  
380 so that IFS and TM5\_co2 run at the same speed while NEMO is a bit faster. Note that  
381 this is also the maximum number of resources we can give to IFS due to the constraints  
382 when running with TM5 described in Section 4.2 As discussed in Section 4.3, we have  
383 chosen to spread LPJG processes so that we can reduce the number of resources needed  
384 from 144 processes (3 full nodes) to only 20. Note that we have reduced a bit more the  
385 number of processes used for LPJG even though this increases the initialization phase  
386 time. In this case, however, the number of PEs that will remain IDLE during that time  
387 is less than in the EC-Earth-Veg case as we are using fewer cores for IFS and NEMO (due  
388 to the TM5.CO2 being slower and limiting the maximum of IFS cores). The results ob-  
389 tained with this new setup are summarized in Table 4. The Coupling cost has been re-  
390 duced by half, the CHSY has improved by 32% (we use less PEs) and the coupled SYPD  
391 is 9% better.

## 392 5 Future work and Conclusions

393 Achieving the best performance of coupled Earth System Models (ESMs) is impos-  
394 sible without studying the scalability properties of their constituents and how they are  
395 linked during the simulation. Without the right tools and metrics needed to understand  
396 the behaviour of these complex applications and a well-grounded methodology, we per-  
397 form Earth System simulations without using the HPC resources effectively due to load-  
398 balance issues.

399 In this paper, we have presented which performance metrics are required and how  
400 to interpret them in order to balance different Coupled Model Intercomparison Project  
401 Phase 6 (CMIP6) configurations of EC-Earth3 that couple up to 4 different components.  
402 Furthermore, we have introduced a new metric (Component\_cpl\_cost) which helps to iden-  
403 tify which of the multiple coupled components is the bottleneck of the ESM execution.  
404 During our analysis, we have shown that intuitive approaches like running all the con-  
405 stituents at the same speed may lead to suboptimal configurations, we have encountered  
406 components that barely scale and limit the speed of the whole coupled model and com-  
407 ponents that need extra resources due to their memory requirements. All in all, we have  
408 been able to identify all these problems and successfully found new resource setups fol-  
409 lowing a new methodology for all the configurations under study that are better in time  
410 and/or energy compared to the previous ones used by the community at the Barcelona  
411 Supercomputing Center.

412 In the future, we are expecting ESMs to grow in complexity and in the number of  
413 constituents that they will include. Performing these load-balance studies will be key to  
414 make the best possible usage of the current and new HPC platforms that are to come.  
415 However, the work of manually finding the best resource setup for all the possible con-

416 configurations of an ESM is very time-consuming and not affordable for many of the teams  
 417 whose main focus is on the Earth’s science, as any change in the model (e.g. components  
 418 used, grid resolution, output intensity, compilation flags, coupling configuration, etc.)  
 419 may require to repeat the analysis and tweak the resources used for each particular case.  
 420 Therefore, we believe that it would be essential to create a tool that can automatically  
 421 balance any ESM, finding the optimal number of resources to use for any number of cou-  
 422 pled components depending on the particular needs of the scientists and bearing in mind  
 423 the existing HPC platform constraints (e.g. time vs. energy solutions, queue limitations  
 424 on the wall-clock or on the maximum number of cores, etc.).

## 425 Open Research Section

426 The scalability plots and data for stand-alone executions of EC-Earth3 components  
 427 can be found in the following GitLab repository:

428 <https://earth.bsc.es/gitlab/spalomas/ec-earth3-scalability-analysis>.

429 The sources for EC-Earth3 ESM can be found on the main web page:

430 <https://ec-earth.org/>.

431 Bear in mind that due to IFS code licence of ECMWF, the development portal (SVN  
 432 repository) can only be accessed by the EC-Earth consortium.

433 Finally, the OASIS-MCT3 coupler sources can be found on their main GitHub page:

434 [https://gitlab.com/cerfacs/oasis3-mct/-/tree/OASIS3-MCT\\_3.1](https://gitlab.com/cerfacs/oasis3-mct/-/tree/OASIS3-MCT_3.1).

## 435 Acknowledgments

436 This project has been a lot of work, but we couldn’t have done it without the support  
 437 and guidance from some very important people. We want to thank Dr María Gonçalves  
 438 and Mr Miguel Castrillo for all their help with this project; they provided us with re-  
 439 sources as well as essential information that was needed to complete our task success-  
 440 fully.

441 We also want to thank all of the people within the Computational Earth Science  
 442 group at the BSC for working alongside us on this project and providing the crucial tools  
 443 to achieve our goals.

## 444 References

- 445 Acosta, M., Yepes-Arbós, X., Valeke, S., Maisonnave, E., Serradell, K., Mula-  
 446 Valls, O., & Doblas-Reyes, F. (2016). *Performance analysis of ec-earth*  
 447 *3.2: Coupling*. Retrieved from [https://earth.bsc.es/wiki/lib/exe/](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)  
 448 [fetch.php?media=library:external:technical\\_memoranda:bsc-ces-2016](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)  
 449 [-006-coupling\\_ec-earth.pdf](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)
- 450 Balaji, V. (2015). Climate computing: The state of play. *Computing in Science &*  
 451 *Engineering*, *17*, 9-13. doi: 10.1109/MCSE.2015.109
- 452 Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U.,  
 453 ... Wright, G. (2017). Cpmip: measurements of real computational perfor-  
 454 mance of earth system models in cmip6. *Geoscientific Model Development*, *10*,  
 455 19-34. Retrieved from <https://gmd.copernicus.org/articles/10/19/2017/>  
 456 doi: 10.5194/gmd-10-19-2017
- 457 Donners, J., Basu, C., Mckinstry, A., Asif, M., Porter, A., Maisonnave, E., ...  
 458 Fladrich, U. (2012, 6). Performance analysis of ec-earth 3.1. *PRACE Whitepa-*  
 459 *per*.
- 460 Döscher, R., Acosta, M., Alessandri, A., Anthoni, P., Arsouze, T., Bergman, T., ...  
 461 Zhang, Q. (2022). The ec-earth3 earth system model for the coupled model  
 462 intercomparison project 6. *Geoscientific Model Development*, *15*, 2973-3020.

- 463 Retrieved from <https://gmd.copernicus.org/articles/15/2973/2022/>  
 464 doi: 10.5194/gmd-15-2973-2022
- 465 Hanke, M., Redler, R., Holfeld, T., & Yastremsky, M. (2016, 8). Yac 1.2.0: new  
 466 aspects for coupling software in earth system modelling. *Geoscientific Model*  
 467 *Development*, 9, 2755-2769. Retrieved from [https://gmd.copernicus.org/](https://gmd.copernicus.org/articles/9/2755/2016/)  
 468 [articles/9/2755/2016/](https://gmd.copernicus.org/articles/9/2755/2016/) doi: 10.5194/gmd-9-2755-2016
- 469 Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., ... Wang, L. (2014,  
 470 10). C-coupler1: a chinese community coupler for earth system model-  
 471 ing. *Geoscientific Model Development*, 7, 2281-2302. Retrieved from  
 472 <https://gmd.copernicus.org/articles/7/2281/2014/> doi: 10.5194/  
 473 gmd-7-2281-2014
- 474 Maisonnave, E., Coquart, L., & Piacentini, A. (2020). *A better diagnostic of the*  
 475 *load imbalance in oasis based coupled systems* (Tech. Rep.). Technical Report,  
 476 TR/CMGC/20/176, CECI, UMR CERFACS/CNRS No5318, France.
- 477 Valcke, S. (2013). The oasis3 coupler: a european climate modelling community soft-  
 478 ware. *Geoscientific Model Development*, 6, 373-388. Retrieved from [https://](https://gmd.copernicus.org/articles/6/373/2013/)  
 479 [gmd.copernicus.org/articles/6/373/2013/](https://gmd.copernicus.org/articles/6/373/2013/) doi: 10.5194/gmd-6-373-2013
- 480 Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., ... Verten-  
 481 stein, M. (2012, 12). Coupling technologies for earth system mod-  
 482 elling. *Geoscientific Model Development*, 5, 1589-1596. Retrieved from  
 483 <https://gmd.copernicus.org/articles/5/1589/2012/> doi: 10.5194/  
 484 gmd-5-1589-2012
- 485 Will, A., Akhtar, N., Brauch, J., Breil, M., Davin, E., Ho-Hagemann, H. T. M., ...  
 486 Weiher, S. (2017, 4). The cosmo-clm 4.8 regional climate model coupled to  
 487 regional ocean, land surface and global earth system models using oasis3-mct:  
 488 description and performance. *Geoscientific Model Development*, 10, 1549-1586.  
 489 Retrieved from <https://gmd.copernicus.org/articles/10/1549/2017/>  
 490 doi: 10.5194/gmd-10-1549-2017

1

# Balancing EC-Earth3

2

## Improving the performance of EC-Earth CMIP6 configurations by minimizing the coupling cost

3

4

M. C. Acosta

5

S. Palomas

6

E. Tourigny

7

Barcelona Supercomputing Center

8

### Key Points:

9

- Find what are the most useful performance metrics and how to understand them in order to tackle the load-imbalance problem in coupled ESMs

10

11

- Understand the individual scalability properties of the components and their relationships in multiple EC-Earth3 CMIP6 configurations

12

13

- Contrast our approach against simpler and traditional ones that the community use to deal with the load-balance problem

14

---

Corresponding author: M. C. Acosta, [mario.acosta@bsc.es](mailto:mario.acosta@bsc.es)

Corresponding author: S. Palomas, [sergi.palomas@bsc.es](mailto:sergi.palomas@bsc.es)

Corresponding author: E. Tourigny, [etienne.tourigny@bsc.es](mailto:etienne.tourigny@bsc.es)

**Abstract**

Earth System Models (ESMs) are complex systems used in weather and climate studies generally built from different independent components responsible for simulating a specific realm (ocean, atmosphere, biosphere, etc.). To replicate the interactions between these processes, ESMs typically use coupling libraries that manage the synchronization and field exchanges between the individual components, which run in parallel as a Multi-Program, Multiple-Data (MPMD) application. As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in HPC platforms has become increasingly challenging and of major concern. One of the critical bottlenecks is the load-imbalance, where the fastest components will have to wait for the slower ones. Finding the optimal number of processing elements (PEs) to assign to each of the multiple independent constituents to minimize the performance loss due to synchronizations and maximize the overall parallel efficiency is impossible without the right performance metrics, methodology and tools. This paper presents the results of balancing multiple Coupled Model Intercomparison Project phase 6 (CMIP6) configurations for the EC-Earth3 ESM. We will show that intuitive approaches can lead to suboptimal resource allocations and propose new setups up to 25% faster while reducing the computational cost by 72%. We prove that new methods are needed to deal with the load-balance of ESMs and hope that our study will serve as a guide to optimize any other coupled system.

**Plain Language Summary**

Earth System Models (ESMs) are complex systems used in weather and climate studies generally built from different independent components responsible for simulating a specific realm (ocean, atmosphere, biosphere, etc.). To replicate the interactions between these processes, ESMs communicate during the simulation to exchange data. As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in HPC platforms has become increasingly challenging and of major concern. One of the critical bottlenecks is the load-imbalance, where the fastest components will have to wait for the slower ones. Finding the optimal number of processing elements (PEs) to assign to each of the independent components without the right performance metrics, methodology and tools. We will show that intuitive approaches can lead to suboptimal setups and propose new ones up to 25% faster while reducing the computational cost by 72%. Thus, proving that new methods are needed to deal with the load-balance of ESMs and hope that our study will serve as a guide to optimize any other coupled system.

## 50 1 Introduction

51 The load-balance of Earth System Models (ESMs), where different components (ocean,  
52 atmosphere, land, sea ice, etc.) are running concurrently, is increasingly complex as we  
53 keep introducing more features during the simulation. Although some models run the  
54 different components in sequential mode, the most common case in the community is the  
55 execution of the different components in parallel (separate cores) and using a coupler to  
56 synchronize the components and exchange information among them. These kinds of ap-  
57 plications are computationally known as MPMD (Multiple Program, Multiple Data), where  
58 different binaries are executed in parallel simulating a natural phenomena using a par-  
59 allel paradigm such as MPI. Given the nature of the physics underneath, the components  
60 within the system have to interact during the simulation (i.e. coupled). Running cou-  
61 pled ESM adds significant changes in the computational performance:

- 62 • Coupling data must be interpolated (regridded), adding extra computation com-  
63 pared to standalone runs. Furthermore, interpolation constraints such as conser-  
64 vation could require serialization techniques and may reduce the parallel efficiency  
65 of the model
- 66 • Coordination among components is needed to exchange data between them. De-  
67 pending on the setup, faster components will have to wait for the slower ones, re-  
68 sulting in IDLE processes and an overall reduction of the parallel efficiency
- 69 • The speed (i.e. parallelization) at which each independent has to run to minimize  
70 the cost of the synchronizations will depend on the coupled configuration. Com-  
71 ponents can no longer run at their optimal scalability point but rather at the op-  
72 timal point for the whole system

73 As we will show, the waiting time due to the synchronization between multiple com-  
74 ponents in a coupled ESM can have a negative effect on the performance achieved. We  
75 have observed that it can consume up to 75% of the total computational cost of the sim-  
76 ulation. In this work, we present the methodology used and the results obtained to bal-  
77 ance different Coupled Model Intercomparison Project phase 6 (CMIP6) configurations  
78 of EC-Earth3, each one having its own particularities (irregular timesteps, memory re-  
79 quirements, different scalability properties, processor mapping, etc.). The solutions achieved  
80 to obtain the best possible Processing Elements (PEs) setup will be contrasted with tra-  
81 ditional methods that are often adopted but, as we will see, can lead to a waste of com-  
82 putational resources. Furthermore, as we show how to optimize setups under distinct con-  
83 texts by exploring multiple configurations of EC-Earth3, we hope that the results will  
84 help with new load-balancing studies of other ESMs, as we have examined common pat-  
85 terns of this type of applications. To achieve our goals, we needed to extend the current  
86 set of metrics in the Computational Performance for Model Intercomparison Projects  
87 (Balaji et al., 2017) (CPMIP) and tools to get them.

## 88 2 Related work

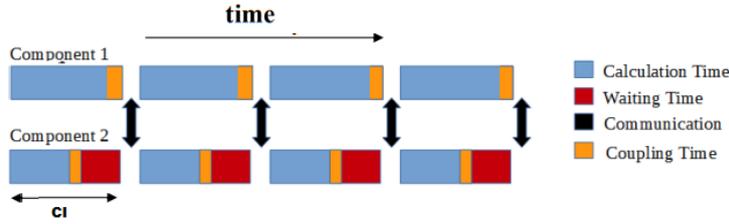
89 Models that simulate the Earth’s climate are among the most computationally-intensive  
90 applications that run on HPC platforms nowadays. Still, the performance that these mod-  
91 els achieve is far from ideal as shown by Balaji (2015), and one of the main limiting fac-  
92 tors is the load-imbalance. Valcke et al. (2012) have shown that many of the current cli-  
93 mate applications used in several institutions are built from different individual compo-  
94 nents and their interactions are managed by a coupler. Although different approaches  
95 exist to couple the components in an ESM, one of the most commonly used is to keep  
96 each component as a separate binary and use the coupling library API calls to transform  
97 and exchange the fields during the simulation. The coupling library ensures the synchro-  
98 nization and regridding processes. Some notable examples are the OASIS3-MCT (Valcke,  
99 2013), C-Coupler (Liu et al., 2014) and YAC (Hanke et al., 2016) couplers. EC-Earth3

100 (Döscher et al., 2022) is a very well-known ESM used in many European institutions and  
 101 uses the OASIS3-MCT coupling library. During the simulation, different components ex-  
 102 change fields and they must be synchronized, rising the load-imbalance problem. The  
 103 fastest components will have to wait for the slower ones to finish before sending/receiving  
 104 the data. The process of finding the best number of PEs to assign to each one of the com-  
 105 ponents which minimizes the overall performance loss due to the synchronization among  
 106 multiple binaries is known as balancing a coupled ESM. An example of dealing with the  
 107 load-imbalance has been shown by Will et al. (2017) for the COSMO-CLM regional cli-  
 108 mate model. Like in EC-Earth3, this ESM uses OASIS3-MCT to couple the multiple bi-  
 109 naries (atmosphere, ocean, etc.) and they used the LUCIA tool (Maisonave et al., 2020)  
 110 to find the optimal number of processes for each component considering the simulation  
 111 time, energy cost and parallel efficiency of the simulation. They stated that the coupling  
 112 time will be minimum if one finds an allocation in which all components run at the same  
 113 speed. The approach consisted of 1) finding a setup in which all components run at the  
 114 same speed with few resources, 2) doubling the number of PEs assigned to each com-  
 115 ponent, 3) readjusting the PEs given to each component so that they run at the same  
 116 speed again, 4) loop to 2 if none of the components' parallel efficiency is below 50%. Even  
 117 though this approach is simple, intuitive and the most frequently used by the commu-  
 118 nity, we will show that it can lead to suboptimal setups. Donners et al. (2012) showed  
 119 that the coupling overhead was an important limiting factor of EC-Earth3 performance.  
 120 They found out that the results obtained from the LUCIA tool could be misleading and  
 121 the approach of running the ocean and atmospheric components at the same speed was  
 122 not good enough to reduce the coupling cost to the minimum. Moreover, they also stud-  
 123 ied in (Acosta et al., 2016) the computing cost of using conservative remapping algorithms  
 124 in the coupler.

125 To analyse the performance of ESMs and evaluate the overhead due to the coupling,  
 126 we will need the right set of performance metrics. As noted by Balaji et al. (2017), given  
 127 the heterogeneity of HPC platforms on which these models run, the differences between  
 128 multiple implementations of ESMs and the varying configurations that can be used, typ-  
 129 ical performance metrics like the FLOPS, cache miss ratio, etc. may not be sufficient for  
 130 the whole range of ESMs. This led to the proposal of the Computational Performance  
 131 Model Intercomparison Project (CPMIP) metrics, which are a collection of metrics es-  
 132 pecially designed for ESMs. In this article, we will use and extend some of them to ac-  
 133 curately address the load-imbalance problem.

### 134 **3 Coupled ESMs**

135 One of the most used approaches to couple ESMs is to keep each individual com-  
 136 ponent as an independent code and use a coupling library (such as OASIS) that deals  
 137 with all the communication between them. This coupling approach is referred to as us-  
 138 ing an "external coupler or coupling library". While it offers the advantage that the changes  
 139 in the source code of each component needed to build the coupled ESM are minimum  
 140 (i.e. keeping independently developed codes self-contained) this implementation has some  
 141 drawbacks to the performance achieved: 1) components will run concurrently on sepa-  
 142 rate PEs and will have to send the exchanged fields across different nodes through the  
 143 HPC network, 2) dependencies between components will reduce the parallel efficiency  
 144 of the ESM as the fastest ones will have to wait for the slowest, 3) an extra computa-  
 145 tion may be needed to transform the data from one component grid to another before  
 146 sending the coupled field. Figure 1 shows the common coupling pattern between two com-  
 147 ponents using an external library. Reducing the IDLE time due to the synchronization  
 148 between components is of utmost importance to achieve a well-balanced ESM and use  
 149 the HPC resources effectively.



**Figure 1.** Overview of two independent components using a coupling library to build an ESM. Each component runs on separate PEs. At the end of each coupling interval (CI), both components need to exchange some coupling fields. The calculation time of Component 2 is less (in blue) and has to wait (in red) for Component 1, which is slower. Furthermore, the execution of both components is extended due to interpolation (in orange) and some fields are exchanged before starting the next CI (black arrows)

150

### 3.1 EC-Earth3 coupling configurations

151

152

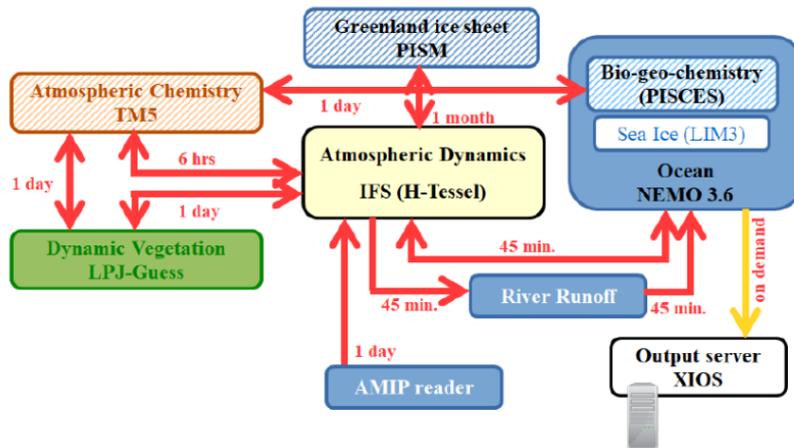
153

154

155

156

The ESM for which we have conducted the load-balance studies is EC-Earth3 (Döscher et al., 2022) which was used in the Coupled Model Intercomparison Project phase 6 (CMIP6) project. The EC-Earth Consortium brings together 27 research institutes from 10 European countries to collaborate on the development of the EC-Earth3 ESM. EC-Earth3 is a fully coupled Atmosphere-Ocean-Land-Biosphere model, that can be used in seasonal to decadal predictions and climate change projections.



**Figure 2.** Overview of EC-Earth3 with the coupling links between all components that can be coupled (Döscher et al., 2022).

157

158

159

160

161

162

As shown in Figure 2, there are multiple possible configurations in EC-Earth3 depending on the individual components used, which are: IFS for the atmosphere and land surface; NEMO for the ocean, sea ice (LIM3 module) and biogeochemistry (PISCES module); LPJ-GUESS (hereafter named LPJG) for the dynamic vegetation; and TM5 for the Atmospheric composition. Each component can run in standalone mode and uses different grids and input data. During this work, we have studied the following 4 different

163 EC-Earth3 coupled configurations where these components run in parallel using separate processors:  
164

- 165 • EC-Earth3: IFS + NEMO
- 166 • EC-Earth3-Veg: IFS + NEMO + LPJG
- 167 • EC-Earth3-AerChem: IFS + NEMO + TM5
- 168 • EC-Earth3-CC: IFS + NEMO + LPJG + TM5\_CO2 + PISCES

169 The resolution used for these configurations is TL255-ORCA1, corresponding to  $\tilde{80}$ km  
170 for the atmosphere,  $1^{\circ}$  for the ocean, with a coupling and component timestep frequency  
171 of 2700s (45min) for IFS and NEMO. All the coupling process is handled by the OASIS3-  
172 MCT coupling library. In addition, NEMO uses the XIOS library to allow having multiple  
173 IO servers that manage the output independently from the model processors and  
174 there is the RiverRunoff process that collects surface and sub-surface runoff from IFS  
175 and eventually sends this as runoff to NEMO. None of them are significant for the load-  
176 balance and we chose to include them in our analysis. For more information about these  
177 configurations, please refer to Döscher et al. (2022).

### 178 3.2 Environment

179 All simulations have been executed in the Barcelona Supercomputing Center HPC  
180 machine MareNostrum4, using Intel Xeon Platinum 8160 processors from the Skylake  
181 generation. It is a Lenovo system composed of SD530 Compute Racks, an Intel Omni-  
182 Path high-performance network interconnect and running SuSE Linux Enterprise Server  
183 as the operating system. Its current Linpack Rmax Performance is 6.23 Petaflops. This  
184 general-purpose block consists of 48 racks housing 3456 nodes with 48 PEs each. Giving  
185 a grand total of  $3456 * 48 = 165,888$  processor cores and 390 Terabytes of main  
186 memory.

### 187 3.3 Performance metrics

188 CPMIP (Balaji et al., 2017) are a collection of performance metrics used to evaluate  
189 the performance of ESMs. The ones used in this work are:

- 190 • Runtime (T): The total execution time of the run
- 191 • Parallelization (P): The number of PEs allocated for the run
- 192 • SYPD: The number of Simulated Years per Day (24 hours of executing time on  
193 the HPC platform)
- 194 • CHSY: The number of Core-Hours per Simulated Year
- 195 • Coupling cost: The fractional cost associated with the coupling events. This in-  
196 cludes the time waiting, sending and interpolating the data.

$$197 \quad Cpl\_cost = \frac{TP - \sum_c T_C P_C}{TP} \quad (1)$$

197 Where  $T_C$  and  $P_C$  are the runtime and parallelization of each component.

198 Additionally, we have introduced the component coupling cost (Component\_cpl\_cost),  
199 which measures how much each component adds to the overall  $Cpl\_cost$ .

$$200 \quad Component\_cpl\_cost = \frac{T_{C_{cpl}} P_C}{TP} \quad (2)$$

200 The OASIS3-MCT coupling library will record the starting and ending times of each cou-  
201 pling event (waiting, sending, interpolating). After the run, this timing information is  
202 post-processed using the LUCIA (Maisonave et al., 2020) tool to collect the mentioned  
203 CPMIP metrics for the simulation.

## 204 4 Results

205 In this section, we show the results obtained when balancing multiple EC-Earth3  
 206 configurations currently used for various climate studies. Firstly, we will discuss two dif-  
 207 ferent solutions for the most common experiment, consisting of IFS coupled with NEMO  
 208 (EC-Earth3 SR). Then we will analyze how introducing the TM5 component in the EC-  
 209 Earth3-AerChem configuration limits the coupled model scaling. Finally, we show which  
 210 is the best approach to allocate the LPJG processes in Carbon-cycle experiments, includ-  
 211 ing the modifications in the total number of processes in EC-Earth3-Veg and EC-Earth3-  
 212 CC configurations. For the results, we have used a high-priority queue. Although it re-  
 213 duces queuing time by having near-instant access to the HPC resources, it imposes two  
 214 constraints: The maximum number of concurrent resources for a job and user is limited  
 215 to 768 PEs ( $768/48 = 16$  nodes) and the wall-clock time is limited to 2h. We include  
 216 this information as relevant to prove that restrictions such as parallel resources or job  
 217 max duration can be managed with the proper methodology.

### 218 4.1 EC-Earth3 SR: IFS-NEMO

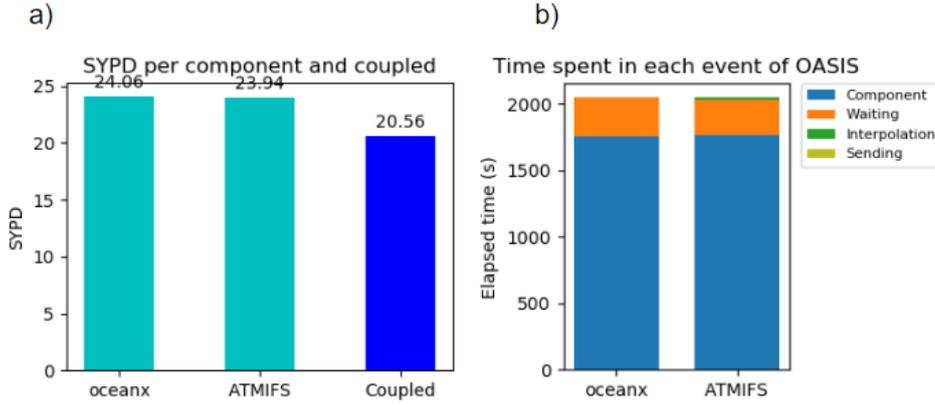
219 In this part, we will compare the typical approach used for the community (same  
 220 SYPD) versus the new one proposed in this work.

#### 221 4.1.1 Same SYPD approach

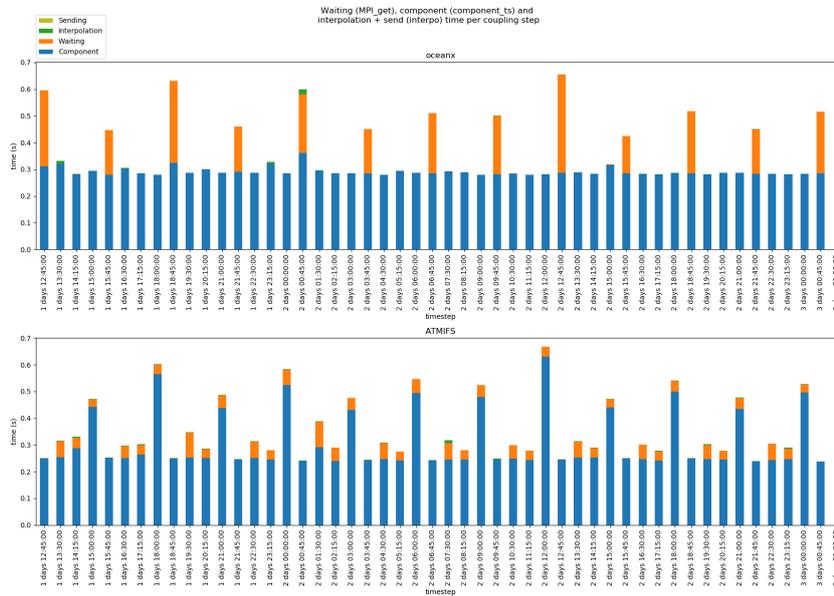
222 A common approach consists of finding a configuration in which all components  
 223 run at the same speed (i.e. SYPD). If we can achieve this, the waiting time due to model  
 224 synchronizations would ideally be 0. Figure 3 shows a setup for which IFS and NEMO  
 225 run at the same SYPD by using 11 ( $48 \cdot 11 = 528$  processes) and 4 nodes ( $48 \cdot 4 = 192$   
 226 processes) for IFS and NEMO, respectively. This is the fastest possible configuration where  
 227 both components' SYPD are similar and fit into the 16-node high-priority queue limita-  
 228 tion (15 for both components and 1 reserved for XIOS). As we see, however, the cou-  
 229 pled SYPD is much lower than the one expected, which theoretically should be approx-  
 230 imately as fast as the lowest SYPD achieved by each independent component.

231 Figure 3 b) shows the time spent for the component execution, waiting, interpo-  
 232 lation and sending for IFS and NEMO for this configuration, obtained through the LU-  
 233 CIA tool. It shows that both components take approximately the same time to finish  
 234 their own execution (in blue) as expected, given that both run at the same SYPD. But  
 235 surprisingly, both components' coupled execution is extended due to the time lost wait-  
 236 ing. Since those are the only two coupled components in this configuration, this can only  
 237 mean that they are waiting for each other.

238 We can see the detailed coupled information per timestep in Figure 4. NEMO (at  
 239 the top) time-stepping (in blue) is regular during the simulation. Meanwhile, every 3 hours  
 240 of simulation (4 timesteps) IFS (at the bottom) component timestep takes much longer  
 241 than the others. Having a component with irregular timestep lengths is quite common.  
 242 Here we know that is due to IFS computing the radiation but can also happen due to  
 243 IO operations or ice calculation. As we forced both components to run at the same SYPD,  
 244 we have created a cyclical conflict in which every 4 timesteps NEMO will have to wait  
 245 for IFS, while IFS is waiting for NEMO during the other timesteps. Moreover, the wait-  
 246 ing time in the 4th timestep in NEMO equals the sum of the waiting time of the pre-  
 247 vious 3 timesteps in IFS. This pattern is repeated during the whole execution making  
 248 the coupled solution slower than expected (i.e. as fast as the slowest component).



**Figure 3.** Results using the same SYPD strategy in EC-Earth3-SR experiments. a) NEMO (oceanx) and IFS (ATMIFS) components running at the same SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component



**Figure 4.** Timestep component and coupling (Waiting, Interpolation, Sending) times for IFS and NEMO when running at the same SYPD

249

#### 4.1.2 Optimal solution

250

251

252

253

254

255

256

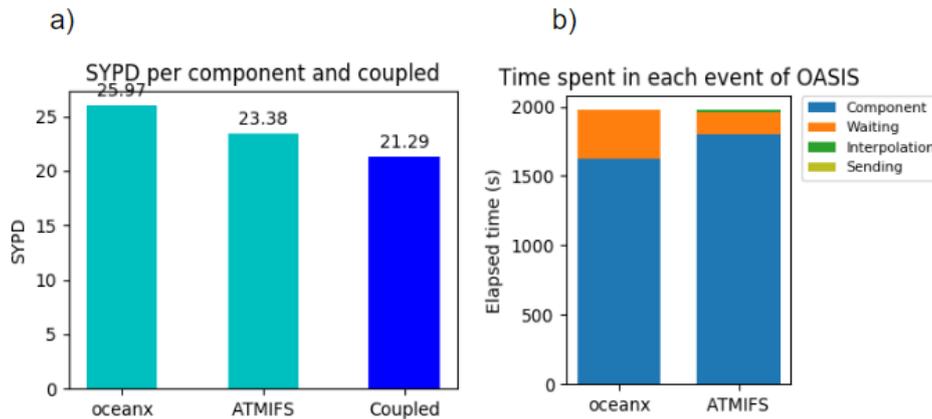
257

As we have seen above, the performance of EC-Earth SR experiments when running IFS and NEMO at the same SYPD seems to be suboptimal since the coupled SYPD is noticeably lower than that of its constituents. As a consequence of forcing both components to have the same computational time, the waiting time due to the coupling synchronizations is also equal (see Figure 3), and the resulting coupling cost is 13.3%. Even though the IFS timestep irregularities observed in Figure 4 imply that it is impossible to find a configuration which reduces the coupling cost to 0, it is still possible to reduce the coupling cost and use the resources more effectively.

**Table 1.** Performance results for EC-Earth-SR experiments when using the same SYPD and our approach

	same SYPD	optimum
SYPD	20.5	21.3
CHSY	896	863
Cpl cost (%)	13.3	10.8
PEs	720	720

258 After running the LUCIA tool, we can take a look at the *Component\_cpl\_cost* met-  
 259 ric (Equation 2) and see that 3.6% of the total coupling cost is due to NEMO waiting  
 260 for IFS and 9.7% due to IFS waiting for NEMO ( $9.7 + 3.6 = 13.3$ ). This means that the  
 261 computing cost of the waiting time is much higher in IFS than in NEMO. Something that  
 262 we were expecting since IFS is using more PEs than NEMO and, even though the wait-  
 263 ing time is the same in both components, more processors are IDLE when IFS has to  
 264 wait. Therefore, it is preferable to give some resources from IFS to NEMO so that IFS  
 265 will wait for less time (NEMO will run faster) and fewer processes will be IDLE. Or in  
 266 other words, we want to reduce the *Component\_coupling\_cost* of the component with  
 267 the highest value for this metric. Moreover, from our single-component scalability anal-  
 268 ysis, we know that NEMO scales better in this range of processor counts. Following this  
 269 approach, we find a new setup with 504 PEs for IFS and 216 for NEMO (we have taken  
 270 24 PEs away from IFS and are now used by NEMO). Thus, the total number of resources  
 271 remains the same but, as we see in Figure 5 a), NEMO is now a bit faster than IFS. With  
 272 this setup, the *Component\_cpl\_cost* of both components is almost the same. This means  
 273 that even though NEMO waiting time is more than twice that of IFS (see Figure 5 b)),  
 274 the cost is the same as it uses fewer PEs. The results achieved are summarized in Ta-  
 275 ble 1. With our analysis, we have found a setup which is 4% faster ( $21.3/20.5$ ) without  
 276 adding any extra resources but only by properly reallocating the PEs from one compo-  
 277 nent to the other using the *Component\_coupling\_cost* metric. As a consequence, the us-  
 278 age of the resources is also better (the CHSY and the coupling cost have also been re-  
 279 duced significantly).

**Figure 5.** Results using an optimal EC-Earth-SR resource configuration. a) NEMO, IFS and Coupled SYPD using 504 and 216 PEs respectively. b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO and IFS using 216 and 504 PEs respectively

280

## 4.2 EC-Earth3-AerChem

281

282

283

284

285

286

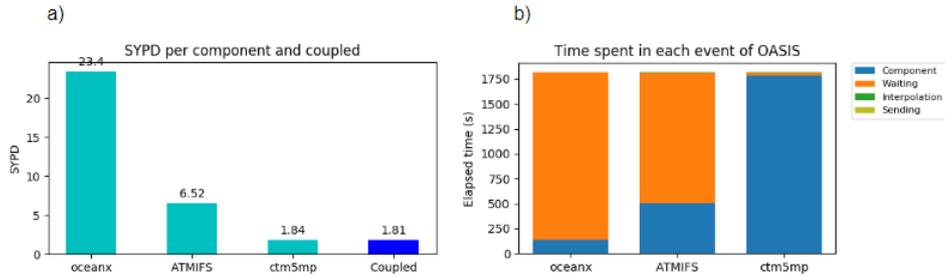
287

288

289

290

The addition of TM5 to EC-Earth in the EC-Earth3-AerChem configuration caused a drastic decrease in the total SYPD that was achieved decreased dramatically. This is mainly because TM5 is very slow compared to the other models and does not scale, being the dominant bottleneck for the coupled simulation as the components have to be synchronized (through the exchange of some particular fields) at each coupling timestep. Moreover, TM5 limits the maximum number of processes that IFS can use to 256 due to the way that spectral fields are exchanged. The default setup previous to this study was using 45 processes for TM5, 256 for IFS and 240 for NEMO. Figure 6 b) shows the magnitude of the overhead introduced by the TM5 component. We also see in Figure 6 a) that this is mainly happening due to IFS and NEMO being much faster than TM5.



**Figure 6.** Results using the default EC-Earth3-AerChem resource configuration. a) NEMO, IFS, TM5 and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO, IFS and TM5 (ctm5mp)

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

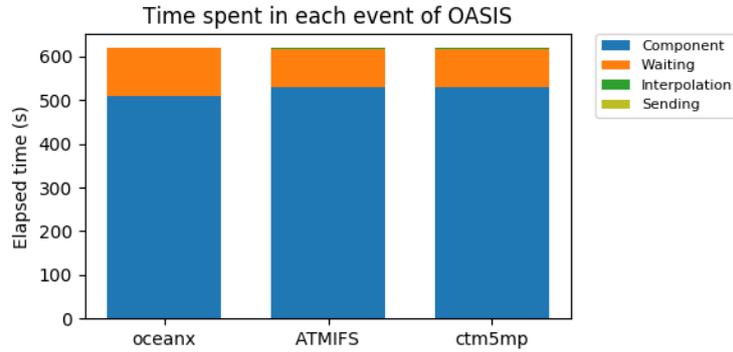
314

315

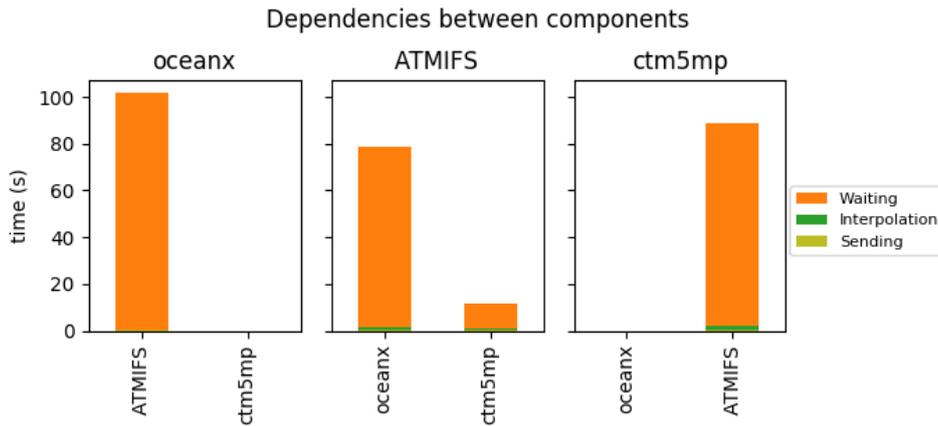
316

The first course of action to optimize the setup for the AerChem configuration was to make TM5 faster, but the scalability tests revealed that this component can barely scale. Nonetheless, we found that it was better to use 90 processes instead of the 45 that were originally allocated, obtaining a speedup of 1.35x. Since we could not further increase the SYPD for this configuration, we decided to save as many cores (and energy) as possible by reducing the number of resources given to the other components, which in this case are IFS and NEMO. Following the same SYPD approach, we end up with a setup that uses 80 PEs for IFS, 16 for NEMO and 90 for TM5, giving a total SYPD of 1.97 and 620 CHSY with a coupling cost of 14.7%. While this configuration is much better than the default one, we did not stop here but rather tried to push the setup a little bit further using our approach with the `Component_cpl_cost`. As we see in Figure 7, we again see that all components' execution time is the same (in blue) but at the same time, they are all waiting during a noticeable amount of time (in orange). Thus, as we saw in Section 4.1, the resulting configuration is less efficient than it could potentially be. By looking at Figure 8 we see that running at the same speed is not only bad for IFS and NEMO, but also for TM5 given that IFS and TM5 are waiting for each other as well. Using this information and the `Component_cpl_cost` metric led us to a configuration with 84 processes for IFS (minimizing the waiting time on TM5 due to IFS), 24 for NEMO (minimizing the waiting time on IFS due to NEMO), and 90 for TM5 (the maximum speed for the slowest component). As explained in Section 4.1.2, we kept NEMO faster than IFS to achieve the best possible combination between these two components while staying just above the SYPD achieved by TM5 (see Figure 9), given that this component is the slowest of the three and we prioritize the speed. The figure also shows that with this new setup, IFS and TM5 spent most of their time without having to wait, while NEMO still suffers from the synchronization but this now only has an effect on 24 PEs, achieving a coupled SYPD of 2.26, 2018 CHSY and 8.25% of coupling cost. Therefore, when

317 comparing with the same SYPD strategy, we managed to make the simulation 1.15x faster  
 318 while reducing the CHSY by 7% and cutting the coupling cost by half.



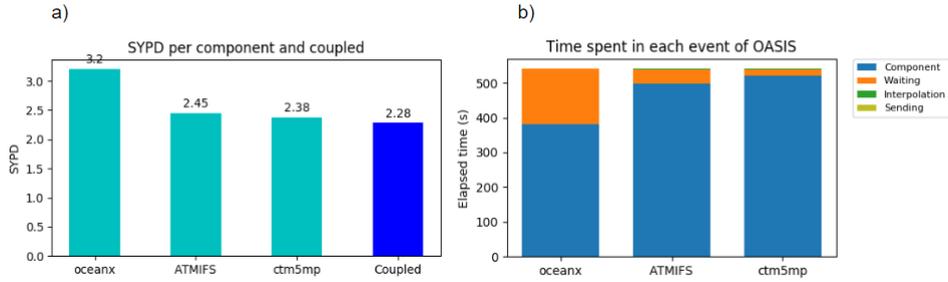
**Figure 7.** Component and coupling (Waiting, Interpolation, Sending) times for an EC-Earth3-AerChem experiment when all components run at the same speed



**Figure 8.** Waiting time between components for an EC-Earth3-AerChem experiment when all components run at the same SYPD. NEMO is always waiting for IFS, IFS is mostly waiting for NEMO and TM5 only waits for IFS

**Table 2.** Performance results of the original, same SYPD and optimal resource setups for EC-Earth-AerChem experiments

	Original	same SYPD	Optimum
SYPD	1.81	1.97	2.26
CHSY	7173	2266	2102
Cpl cost (%)	75.3	14.7	8.25
PEs	541	186	198



**Figure 9.** a) Results using a balanced EC-Earth3-AerChem resource configuration. a) IFS, NEMO, TM5 and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component

319

### 4.3 EC-Earth3-Veg

320

321

322

323

324

325

326

327

This experiment configuration adds LPJG to be coupled with IFS and NEMO. One of the particularities of this component is that it is much faster than those two and, therefore, the strategy of running all components at the same speed can no longer be applied. Figure 10 shows that with the default resource configuration, LPJG spends most of the time waiting, and in Figure 10 we see that this happens because this component is much faster than IFS and NEMO. Ideally, we would like to reduce the number of resources used by LPJG. Still, we found a couple of limitations with this component which have to be taken into account to design an optimal setup for EC-Earth3-Veg configurations:

328

329

330

331

332

333

334

335

336

337

338

339

340

341

- *Memory consumption:* The memory consumption of LPJG is high. On Marens-trum4, it is recommended to use 3 nodes with 96GB of main memory each to ensure that this component won't fail during the simulation due to a lack of memory.
- *Initialization:* Studying the scalability of LPJG we have realized that it is much faster than IFS and NEMO during the execution but it has a slow initialization. We don't need many cores to run LPJG without it interfering with the execution of the other components. However, reducing too much the PEs assigned for LPJG will make the initialization phase slower. This can make hundreds or even thousands of processes (the ones assigned to the other components) wait for the initialization of LPJG at the beginning of the simulation, which can take up to 7 minutes with very few processes. Even though the initialization overhead is mitigated in long simulations, the waste of resources still exists and it could be significant for shorter chunks.

342

343

344

345

346

The only way to reduce the number of PEs used by LPJG while ensuring that it will have access to enough memory is to spread its processes across multiple nodes. Therefore, the use of explicit affinity (to distribute parallel resources through the machine manually) is key to improving this configuration's performance by making it possible to use the memory of multiple nodes, without having to assign all their cores exclusively to LPJG.

347

348

349

To choose whether it is better to share LPJG processes with NEMO or IFS, we have conducted some memory consumption and communication overhead studies for each of these components independently:

350

351

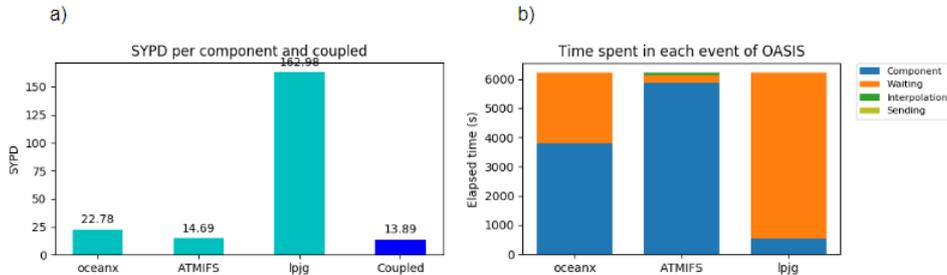
- *The memory consumption:* If IFS or NEMO are consuming too much memory already, LPJG should not share the node with that component. We have tested how

352 the memory consumption of these components changes as we reduce the number  
 353 of cores they use per node.  
 354 • The communication overhead: If the component shows an overhead due to the ex-  
 355 tra communication needed between different nodes after scattering its processes,  
 356 we have to ensure that this loss in efficiency would not be big enough to make the  
 357 explicit affinity solution unworthy.

358 In both cases, IFS and NEMO do benefit from reducing the number of cores per  
 359 node they use and the MPI communication overhead is negligible. According to the re-  
 360 sults obtained, the memory consumption of IFS is higher and we concluded that it is bet-  
 361 ter to make NEMO and LPJG components share resources. The optimal setup uses 336  
 362 processes for IFS, 380 for NEMO and 40 for LPJG. A total of 8 nodes are used by NEMO  
 363 and LPJG at the same time, the first running on 43 cores and the latter on the remain-  
 364 ing 5. Not only is this configuration more efficient, but it also now fits into the debug  
 365 queue as it uses less than 16 nodes and the time to solution for 1-year simulations is less  
 366 than 2h. The results are shown in Table 3. Although the coupled SYPD achieved is 5%  
 367 (13.2/13.9) lower, the number of resources needed has decreased by 40% (768/1104) and  
 368 the CHSY is now 28% (1318/1824) better.

**Table 3.** Performance results of the original and optimal resource setup for EC-Earth-Veg experiments

	Original	Optimum
SYPD	13.9	13.2
CHSY	1824	1318
Cpl cost (%)	28.7	21
PEs	1104	768



**Figure 10.** a) Results using the default EC-Earth3-Veg resource configuration. a) NEMO, IFS, LPJG and coupled SYPD. b) Component and coupling (Waiting, Interpolation, Sending) times per component

#### 369 4.4 EC-Earth3-CC

370 The last of the configurations to evaluate consists again of IFS, NEMO and LPJG  
 371 but it also adds a reduced version of TM5 to simulate the atmospheric Carbon cycle (TM5\_CO2).  
 372 Again, with LPJG we can not use the same SYPD strategy. Instead, we will again show  
 373 how spreading the physical allocation of its processes is the best approach to minimize  
 374 the performance loss by this component and how to balance an experiment with 4 cou-  
 375 pled components. TM5\_CO2 does not scale very well, but after doing the scalability anal-

**Table 4.** Performance results of the original and optimal resource setup for EC-Earth-CC experiments

	Original	Optimum
SYPD	7.1	7.73
CHSY	2104	1428
Cpl cost (%)	33.7	15.1
PEs	621	476

376 ysis we found that it is much faster than the full TM5 execution and that instead of us-  
377 ing 45 processes (as the default resource configuration suggested), this component is faster  
378 when using only 8 processes, achieving almost 9 SYPD. Given that TM5 limits the ex-  
379 ecution speed, we have reduced IFS and NEMO processes to 256 and 192 respectively,  
380 so that IFS and TM5\_co2 run at the same speed while NEMO is a bit faster. Note that  
381 this is also the maximum number of resources we can give to IFS due to the constraints  
382 when running with TM5 described in Section 4.2 As discussed in Section 4.3, we have  
383 chosen to spread LPJG processes so that we can reduce the number of resources needed  
384 from 144 processes (3 full nodes) to only 20. Note that we have reduced a bit more the  
385 number of processes used for LPJG even though this increases the initialization phase  
386 time. In this case, however, the number of PEs that will remain IDLE during that time  
387 is less than in the EC-Earth-Veg case as we are using fewer cores for IFS and NEMO (due  
388 to the TM5.CO2 being slower and limiting the maximum of IFS cores). The results ob-  
389 tained with this new setup are summarized in Table 4. The Coupling cost has been re-  
390 duced by half, the CHSY has improved by 32% (we use less PEs) and the coupled SYPD  
391 is 9% better.

## 392 5 Future work and Conclusions

393 Achieving the best performance of coupled Earth System Models (ESMs) is impos-  
394 sible without studying the scalability properties of their constituents and how they are  
395 linked during the simulation. Without the right tools and metrics needed to understand  
396 the behaviour of these complex applications and a well-grounded methodology, we per-  
397 form Earth System simulations without using the HPC resources effectively due to load-  
398 balance issues.

399 In this paper, we have presented which performance metrics are required and how  
400 to interpret them in order to balance different Coupled Model Intercomparison Project  
401 Phase 6 (CMIP6) configurations of EC-Earth3 that couple up to 4 different components.  
402 Furthermore, we have introduced a new metric (Component\_cpl\_cost) which helps to iden-  
403 tify which of the multiple coupled components is the bottleneck of the ESM execution.  
404 During our analysis, we have shown that intuitive approaches like running all the con-  
405 stituents at the same speed may lead to suboptimal configurations, we have encountered  
406 components that barely scale and limit the speed of the whole coupled model and com-  
407 ponents that need extra resources due to their memory requirements. All in all, we have  
408 been able to identify all these problems and successfully found new resource setups fol-  
409 lowing a new methodology for all the configurations under study that are better in time  
410 and/or energy compared to the previous ones used by the community at the Barcelona  
411 Supercomputing Center.

412 In the future, we are expecting ESMs to grow in complexity and in the number of  
413 constituents that they will include. Performing these load-balance studies will be key to  
414 make the best possible usage of the current and new HPC platforms that are to come.  
415 However, the work of manually finding the best resource setup for all the possible con-

416 configurations of an ESM is very time-consuming and not affordable for many of the teams  
 417 whose main focus is on the Earth’s science, as any change in the model (e.g. components  
 418 used, grid resolution, output intensity, compilation flags, coupling configuration, etc.)  
 419 may require to repeat the analysis and tweak the resources used for each particular case.  
 420 Therefore, we believe that it would be essential to create a tool that can automatically  
 421 balance any ESM, finding the optimal number of resources to use for any number of cou-  
 422 pled components depending on the particular needs of the scientists and bearing in mind  
 423 the existing HPC platform constraints (e.g. time vs. energy solutions, queue limitations  
 424 on the wall-clock or on the maximum number of cores, etc.).

## 425 Open Research Section

426 The scalability plots and data for stand-alone executions of EC-Earth3 components  
 427 can be found in the following GitLab repository:

428 <https://earth.bsc.es/gitlab/spalomas/ec-earth3-scalability-analysis>.

429 The sources for EC-Earth3 ESM can be found on the main web page:

430 <https://ec-earth.org/>.

431 Bear in mind that due to IFS code licence of ECMWF, the development portal (SVN  
 432 repository) can only be accessed by the EC-Earth consortium.

433 Finally, the OASIS-MCT3 coupler sources can be found on their main GitHub page:

434 [https://gitlab.com/cerfacs/oasis3-mct/-/tree/OASIS3-MCT\\_3.1](https://gitlab.com/cerfacs/oasis3-mct/-/tree/OASIS3-MCT_3.1).

## 435 Acknowledgments

436 This project has been a lot of work, but we couldn’t have done it without the support  
 437 and guidance from some very important people. We want to thank Dr María Gonçalves  
 438 and Mr Miguel Castrillo for all their help with this project; they provided us with re-  
 439 sources as well as essential information that was needed to complete our task success-  
 440 fully.

441 We also want to thank all of the people within the Computational Earth Science  
 442 group at the BSC for working alongside us on this project and providing the crucial tools  
 443 to achieve our goals.

## 444 References

- 445 Acosta, M., Yepes-Arbós, X., Valeke, S., Maisonnave, E., Serradell, K., Mula-  
 446 Valls, O., & Doblas-Reyes, F. (2016). *Performance analysis of ec-earth*  
 447 *3.2: Coupling*. Retrieved from [https://earth.bsc.es/wiki/lib/exe/](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)  
 448 [fetch.php?media=library:external:technical\\_memoranda:bsc-ces-2016](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)  
 449 [-006-coupling\\_ec-earth.pdf](https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf)
- 450 Balaji, V. (2015). Climate computing: The state of play. *Computing in Science &*  
 451 *Engineering*, *17*, 9-13. doi: 10.1109/MCSE.2015.109
- 452 Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U.,  
 453 ... Wright, G. (2017). Cpmip: measurements of real computational perfor-  
 454 mance of earth system models in cmip6. *Geoscientific Model Development*, *10*,  
 455 19-34. Retrieved from <https://gmd.copernicus.org/articles/10/19/2017/>  
 456 doi: 10.5194/gmd-10-19-2017
- 457 Donners, J., Basu, C., Mckinstry, A., Asif, M., Porter, A., Maisonnave, E., ...  
 458 Fladrich, U. (2012, 6). Performance analysis of ec-earth 3.1. *PRACE Whitepa-*  
 459 *per*.
- 460 Döscher, R., Acosta, M., Alessandri, A., Anthoni, P., Arsouze, T., Bergman, T., ...  
 461 Zhang, Q. (2022). The ec-earth3 earth system model for the coupled model  
 462 intercomparison project 6. *Geoscientific Model Development*, *15*, 2973-3020.

- 463 Retrieved from <https://gmd.copernicus.org/articles/15/2973/2022/>  
 464 doi: 10.5194/gmd-15-2973-2022
- 465 Hanke, M., Redler, R., Holfeld, T., & Yastremsky, M. (2016, 8). Yac 1.2.0: new  
 466 aspects for coupling software in earth system modelling. *Geoscientific Model*  
 467 *Development*, 9, 2755-2769. Retrieved from [https://gmd.copernicus.org/](https://gmd.copernicus.org/articles/9/2755/2016/)  
 468 [articles/9/2755/2016/](https://gmd.copernicus.org/articles/9/2755/2016/) doi: 10.5194/gmd-9-2755-2016
- 469 Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., ... Wang, L. (2014,  
 470 10). C-coupler1: a chinese community coupler for earth system model-  
 471 ing. *Geoscientific Model Development*, 7, 2281-2302. Retrieved from  
 472 <https://gmd.copernicus.org/articles/7/2281/2014/> doi: 10.5194/  
 473 [gmd-7-2281-2014](https://gmd.copernicus.org/articles/7/2281/2014/)
- 474 Maisonnave, E., Coquart, L., & Piacentini, A. (2020). *A better diagnostic of the*  
 475 *load imbalance in oasis based coupled systems* (Tech. Rep.). Technical Report,  
 476 TR/CMGC/20/176, CECI, UMR CERFACS/CNRS No5318, France.
- 477 Valcke, S. (2013). The oasis3 coupler: a european climate modelling community soft-  
 478 ware. *Geoscientific Model Development*, 6, 373-388. Retrieved from [https://](https://gmd.copernicus.org/articles/6/373/2013/)  
 479 [gmd.copernicus.org/articles/6/373/2013/](https://gmd.copernicus.org/articles/6/373/2013/) doi: 10.5194/gmd-6-373-2013
- 480 Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., ... Verten-  
 481 stein, M. (2012, 12). Coupling technologies for earth system mod-  
 482 elling. *Geoscientific Model Development*, 5, 1589-1596. Retrieved from  
 483 <https://gmd.copernicus.org/articles/5/1589/2012/> doi: 10.5194/  
 484 [gmd-5-1589-2012](https://gmd.copernicus.org/articles/5/1589/2012/)
- 485 Will, A., Akhtar, N., Brauch, J., Breil, M., Davin, E., Ho-Hagemann, H. T. M., ...  
 486 Weiher, S. (2017, 4). The cosmo-clm 4.8 regional climate model coupled to  
 487 regional ocean, land surface and global earth system models using oasis3-mct:  
 488 description and performance. *Geoscientific Model Development*, 10, 1549-1586.  
 489 Retrieved from <https://gmd.copernicus.org/articles/10/1549/2017/>  
 490 doi: 10.5194/gmd-10-1549-2017