Earth and Space Science

RESEARCH ARTICLE
10.1029/2023EA002912

Key Points:
- Find what are the most useful performance metrics and how to understand them in order to tackle the load-imbalance problem in coupled Earth System Models
- Understand the individual scalability properties of the components and their relationships in multiple EC-Earth3 CMIP6 configurations
- Contrast our approach against simpler and traditional ones that the community use to deal with the load-balance problem

Correspondence to:
M. C. Acosta, S. Palomas and E. Tourigny,
mario.acosta@bsc.es;
sergi.palomas@bsc.es;
etienne.tourigny@bsc.es

Citation:
Acosta, M. C., Palomas, S., & Tourigny, E. (2023). Balancing EC-Earth3 improving the performance of EC-Earth CMIP6 configurations by minimizing the coupling cost. *Earth and Space Science*, *10*, e2023EA002912. https://doi.org/10.1029/2023EA002912

Received 28 FEB 2023
Accepted 8 JUN 2023

Author Contributions:
Conceptualization: M. C. Acosta
Investigation: S. Palomas
Methodology: M. C. Acosta, S. Palomas
Project Administration: M. C. Acosta
Resources: E. Tourigny
Software: S. Palomas
Supervision: M. C. Acosta, E. Tourigny
Writing – original draft: S. Palomas
Writing – review & editing: S. Palomas, E. Tourigny

Balancing EC-Earth3 Improving the Performance of EC-Earth CMIP6 Configurations by Minimizing the Coupling Cost

M. C. Acosta[1], S. Palomas[1], and E. Tourigny[1]

[1]Barcelona Supercomputing Center, Barcelona, Spain

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 application. As ESMs get more complex (increase in resolution, number of components, configurations, etc.), achieving the best performance when running in High-performance Computing 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 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 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% fasters 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 (ESM) 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 High-performance Computing 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 to assign to each of the independent components is impossible 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% fasters 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 Multiple Program, Multiple Data (MPMD), where different binaries are executed in parallel simulating a natural phenomena using a parallel paradigm such as Message Passing Interface. 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. As seen by Acosta et al. (2021), several institutions reported that the coupling cost accounted for 5%–20% of the total computational cost of the simulations during the Coupled Model Intercomparison Project phase 6 (CMIP6), even though these experiments were carefully tuned to achieve the maximum parallel efficiency, as they were used to simulate hundreds of thousands of years. This highlights the significant impact of the coupling on the overall performance of ESMs, which we estimate accounted for 17 tCO_2 emissions during the CMIP6 runs, emphasizing the need for better strategies to minimize its computational cost.

In this work, we present the methodology used and the results obtained to balance different 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 these types 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. The method we propose is able to find a resource allocation which minimizes the coupling cost given a limitation on the number of processors to use, Max_PEs, and a step, N, for any coupled configuration. It consists of the following procedures: (a) get the scalability properties of the models involved in a coupled ESM configuration, (b) instrument the calls to the coupler to get the time inside and between coupling events for each constituent, (c) run the ESM (one can use the fastest setup following the same SYPD strategy, easily obtained from the scalability curves, or the default used by the community if available), (d) get the computing cost of the coupling events per component (*component_cpl_cost* metric) to identify the bottleneck component, (e) if the Max_PEs restriction has not been met yet, give N PEs to the bottleneck component. Move N resources from the fastest to the bottleneck component otherwise. (f) loop to (c) until the new resource setup does not improve compared to the previous iteration.

Notice that the number of processes to add or move to the bottleneck component (*N*) is ultimately an arbitrary decision, which should be in accordance with the Max_PEs value and slope of the components' scalability properties. Moreover, the method might be adapted for particular cases as we will show in the results section.

2. Related Work

Models that simulate the Earth's climate are among the most computationally-intensive applications that run on High-performance Computing (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 Application Programming Interface 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 (Döscher et al., 2022) is a very well-known ESM used in many European institutions and uses the OASIS3-MCT coupling library. During the simulation, different components exchange fields and they must be synchronized, rising the load-imbalance problem. The fastest components will have to wait for the slower ones to finish before sending/receiving the data. The process of finding the best number of PEs to assign to each one of the components which minimizes the overall performance loss due to the synchronization among multiple binaries is known as balancing a coupled ESM. An example of dealing with the load-imbalance

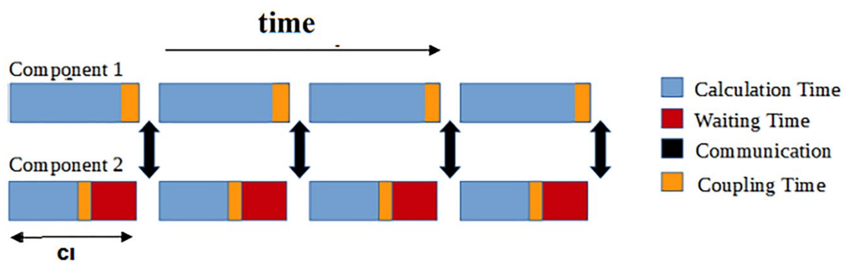

Figure 1. Overview of two independent components using a coupling library to build an Earth System Model. Each component runs on separate processing elements. At the end of each coupling interval (CI), both components need to exchange some coupling fields. The calculation time of Component 2 is less (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).

has been shown by Will et al. (2017) for the Consortium for Small-scale Modeling - Climate Limited-Area Modeling regional climate model. Like in EC-Earth3, this ESM uses OASIS3-MCT to couple the multiple binaries (atmosphere, ocean, etc.) and they used the LUCIA tool (Maisonnave et al., 2020) to find the right number of processes for each component considering the simulation time, energy cost, and parallel efficiency of the simulation. They stated that the coupling time will be minimum if one finds an allocation in which all components run at the same speed. The approach consisted of (a) finding a setup in which all components run at the same speed with few resources, (b) doubling the number of PEs assigned to each component, (c) readjusting the PEs given to each component so that they run at the same speed again, (d) loop to (b) if none of the components' parallel efficiency is below 50%. Even though this approach is simple, intuitive, and the most frequently used by the community, we will show that it can lead to suboptimal setups. Donners et al. (2012) showed that the coupling overhead was an important limiting factor of EC-Earth3 performance. They found out that the results obtained from the LUCIA tool could be misleading and the approach of running the ocean and atmospheric components at the same speed was not good enough to reduce the coupling cost to the minimum. Moreover, they also studied in (Acosta et al., 2016) the computing cost of using conservative remapping algorithms in the coupler.

To analyze the performance of ESMs and evaluate the overhead due to the coupling, we will need the right set of performance metrics. As noted by Balaji et al. (2017), given the heterogeneity of HPC platforms on which these models run, the differences between multiple implementations of ESMs and the varying configurations that can be used, typical performance metrics like the FLOPS, cache miss ratio, etc. may not be sufficient for the whole range of ESMs. This led to the proposal of the Computational Performance Model Intercomparison Project (CPMIP) metrics, which are a collection of metrics especially designed for ESMs. Although we will be using CMIP6 configurations to validate our approach, in this article we will use and extend only the most important ones to address the load-imbalance problem, and compare the results against simulations no longer than 1 year. Note, however, that other metrics which evaluate the cost of system interruptions, IO operations, resolution, complexity, etc. should also be accounted for and longer simulations are needed to properly apprize the benefits over the whole ESM performance, which we hope to include in future works.

3. Coupled ESMs

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

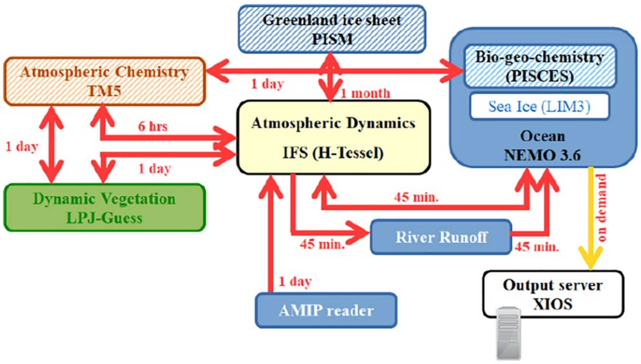

Figure 2. Overview of EC-Earth3 with the coupling links between all components that can be coupled (Döscher et al., 2022).

3.1. EC-Earth3 Coupling Configurations

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.

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 four different EC-Earth3 coupled configurations where these components run in parallel using separate processors:

- EC-Earth3: IFS + NEMO
- EC-Earth3-Veg: IFS + NEMO + LPJG
- EC-Earth3-AerChem: IFS + NEMO + TM5
- EC-Earth3-CC: IFS + NEMO + LPJG + TM5_CO2 + PISCES

The resolution used for these configurations is TL255-ORCA1, corresponding to 80 km for the atmosphere, 1° for the ocean, with a coupling and component timestep frequency of 2700 s (45 min) for IFS and NEMO. All the coupling process is handled by the OASIS3-MCT coupling library. Furthermore, NEMO uses the XIOS library to allow having multiple dedicated IO servers that manage the output independently from the model processors (server mode) and there is the River Runoff process that collects surface and sub-surface runoff from IFS and eventually sends this as runoff to NEMO. While XIOS does not directly interact with the coupling, IO operations can affect the load-balance of the coupled run if they slow down NEMO. Despite this, we did not include XIOS or River Runoff in our analysis, as we already allocated enough XIOS IO servers to mitigate the output cost (Ticco et al., 2020), and the River Runoff process is much faster than any General Circulation Model. For more information about these configurations, please refer to Döscher et al. (2022).

3.2. Environment

All simulations have been executed in the Barcelona Supercomputing Center HPC machine MareNostrum4, using Intel Xeon Platinum 8160 processors from the Skylake generation. It is a Lenovo system composed of SD530 Compute Racks, an Intel Omni-Path high-performance network interconnect and running SuSE Linux Enterprise Server as the operating system. Its current Linpack Rmax Performance is 6.23 Petaflops. This general-purpose block consists of 48 racks housing 3,456 nodes with 48 PEs each. Giving a grand total of 3,456 ∗ 48 = 165,888 processor cores and 390 Terabytes of main memory.

3.3. Performance Metrics

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

- Runtime (T): The total execution time of the run
- Parallelization (P): The number of PEs allocated for the run
- SYPD: The number of Simulated Years per Day (24 hr of executing time on the HPC platform)
- CHSY: The number of Core-Hours per Simulated Year
- Coupling cost: The fractional cost associated with the coupling events. This includes the time waiting, sending, and interpolating the data.

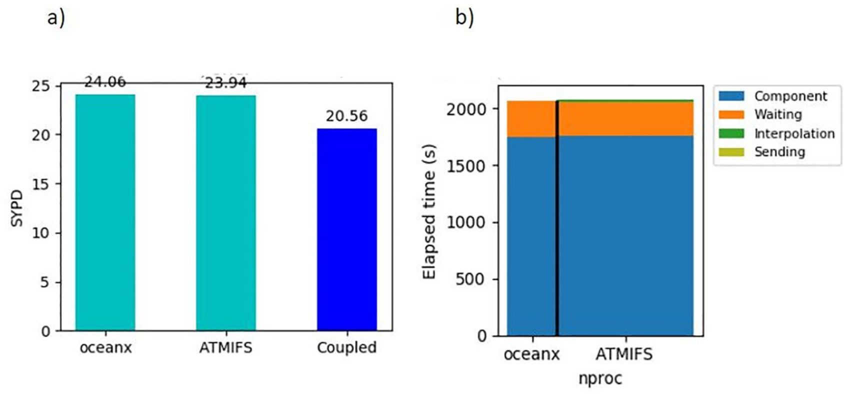

Figure 3. Results using the same SYPD strategy in EC-Earth3-SR experiments. (a) NEMO (oceanx) and IFS (ATMIFS) components running at the same SYPD. (b) Component and coupling (Waiting, Interpolation, Sending) times per component. The width corresponds to the number of processing elements used by each component.

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

where T_C and P_C are the runtime without including the coupling events (i.e., equivalent to a standalone run) and parallelization of each component.

Additionally, we have introduced the component coupling cost (Component_cpl_cost), which measures how much each component adds to the overall *Cpl_cost*.

$$\text{Component\_cpl\_cost} = \frac{T_{Ccpl} P_C}{TP} \quad (2)$$

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

4. Results

In this section, we show the results obtained when balancing multiple EC-Earth3 configurations currently used for various climate studies. First, we will discuss two different solutions for the most common experiment, consisting of IFS coupled with NEMO (EC-Earth3 SR). Then we will analyze how introducing the TM5 component in the EC-Earth3-AerChem configuration limits the coupled model scaling. Finally, we show which is the best approach to allocate the LPJG processes in Carbon-cycle experiments, including the modifications in the total number of processes in EC-Earth3-Veg and EC-Earth3-CC configurations. For the results, we have used a high-priority queue. Although it reduces queuing time by having near-instant access to the HPC resources, it imposes two constraints: The maximum number of concurrent resources for a job and user is 768 PEs (768/48 = 16 nodes) and the wall-clock time is limited to 2 hr.

4.1. EC-Earth3 SR: IFS-NEMO

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

4.1.1. Same SYPD Approach

A common approach consists of finding a configuration in which all components run at the same speed (i.e., SYPD). If we can achieve this, the waiting time due to model synchronizations would ideally be 0. Figure 3 shows a setup for which IFS and NEMO run at the same SYPD by using 11 (48 × 11 = 528 processes) and 4 nodes (48 × 4 = 192 processes) for IFS and NEMO, respectively. This is the fastest possible configuration where both components' SYPD are similar and fit into the 16-node high-priority queue limitation (15 for both components

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

and 1 reserved for XIOS). As we see, however, the coupled SYPD is lower than the one expected, which theoretically should be approximately as fast as the lowest SYPD achieved by each independent component.

Figure 3b shows the time spent for the component execution, waiting, interpolation and sending for IFS and NEMO for this configuration, obtained through the LUCIA tool. It shows that both components take approximately the same time to finish their own execution (in blue) as expected, given that both run at the same SYPD. But surprisingly, both components' coupled execution is extended due to the time lost waiting. Since those are the only two coupled components in this configuration, this can only mean that they are waiting for each other.

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

4.1.2. Balanced Solution

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

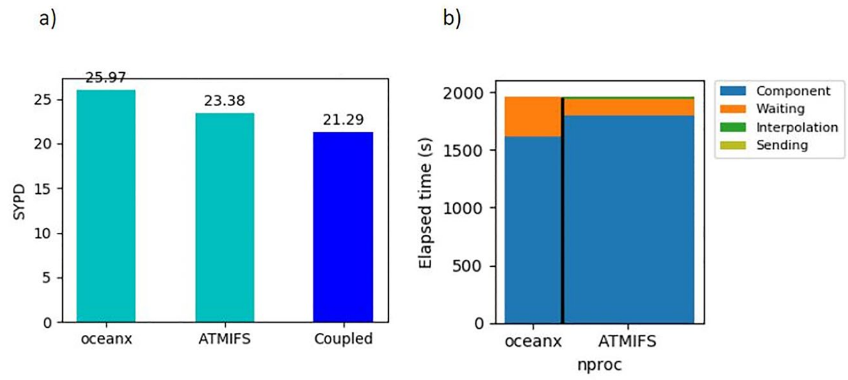

Figure 5. Results using a balanced EC-Earth-SR resource configuration. (a) NEMO, IFS, and Coupled SYPD using 504 and 216 processing elements (PEs) respectively. (b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO and IFS using 216 and 504 PEs respectively. The width corresponds to the number of PEs used by each component.

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.

After running the LUCIA tool, we can take a look at the *Component_cpl_cost* metric (Equation 2) and see that 3.6% of the total coupling cost is due to NEMO waiting for IFS and 9.7% due to IFS waiting for NEMO (9.7 + 3.6 = 13.3). This means that the computing cost of the waiting time is much higher in IFS than in NEMO. Something that we were expecting since IFS is using more PEs than NEMO and, even though the waiting time is the same in both components, more processors are IDLE when IFS has to wait. Therefore, it is preferable to give some resources from IFS to NEMO so that IFS will wait for less time (NEMO will run faster) and fewer processes will be IDLE. Or in other words, we want to reduce the *Component_coupling_cost* of the component with the highest value for this metric. Moreover, from our single-component scalability analysis, we know that NEMO scales better in this range of processor counts. Following this approach, we find a new setup with 504 PEs for IFS and 216 for NEMO (we have taken 24 PEs away from IFS and are now used by NEMO). Thus, the total number of resources remains the same but, as we see in Figure 5a, NEMO is now a bit faster than IFS. With this setup, the *Component_cpl_cost* of both components is almost the same. This means that even though NEMO waiting time is more than twice that of IFS (see Figure 5b)), the cost is the same as it uses fewer PEs. The results achieved are summarized in Table 1. With our analysis, we have found a setup which is 4% faster (21.3/20.5) without adding any extra resources but only by properly reallocating the PEs from one component to the other using the *Component_coupling_cost* metric. As a consequence, the usage of the resources is also better (the CHSY and the coupling cost have also been reduced significantly).

4.2. EC-Earth3-AerChem

The addition of TM5 to EC-Earth in the EC-Earth3-AerChem configuration caused a drastic decrease in the total SYPD that was achieved. 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 6b shows the magnitude of the overhead introduced by the TM5 component. We also see in Figure 6a) that this is mainly happening due to IFS and NEMO being much faster than TM5.

Table 1
Performance Results for EC-Earth-SR Experiments When Using the Same SYPD and Our Approach

	Same SYPD	Balanced
SYPD	20.5	21.3
CHSY	896	863
Cpl cost (%)	13.3	10.8
PEs	720	720

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

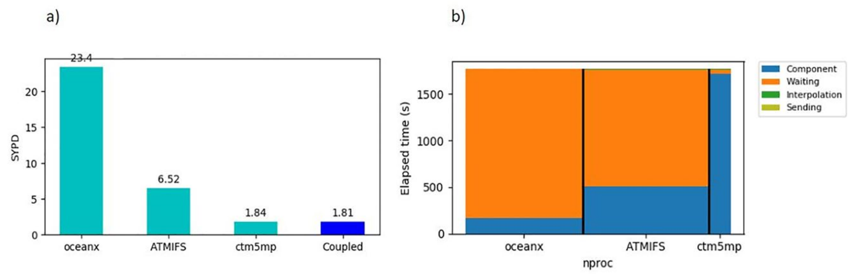

Figure 6. Results using the default EC-Earth3-AerChem resource configuration. (a) NEMO, IFS, TM5, and coupled SYPD. (b) Component and coupling (Waiting, Interpolation, Sending) times for NEMO, IFS, and TM5 (ctm5mp). The width corresponds to the number of processing elements used by each component.

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. As seen in Table 2, 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 the new setup improves the processing time utilization of IFS and TM5, as they almost don't have to wait for synchronization, whereas NEMO still experiences some synchronization issues, but only uses a small pull of PEs (24). As a result, the waiting time cost is reduced. As seen in Table 2, the Balanced setup achieves a coupled SYPD of 2.26, 2018 CHSY, and a coupling cost of 8.25%. When compared with the same SYPD strategy, our optimization made the simulation 1.15x faster, reduced the CHSY by 7%, and halved the coupling cost.

4.3. EC-Earth3-Veg

This experiment configuration adds LPJG to be coupled with IFS and NEMO. One of the particularities of this component is that is much faster than those two and, therefore, the strategy of running all components at the same speed can no longer be applied. Moreover, unlike the other models under study, the performance of the Vegetation model is limited by the available memory and does not scale proportionally to the number of PEs assigned. As a result, the methodology and optimizations had to be adapted specifically for this scenario. Figure 10 shows that

Table 2
Performance Results of the Original, Same SYPD, and Balanced Resource Setups for EC-Earth-AerChem Experiments

	Original	Same SYPD	Balanced
SYPD	1.81	1.97	2.26
CHSY	7,173	2,266	2,102
Cpl cost (%)	75.3	14.7	8.25
PEs	541	186	198

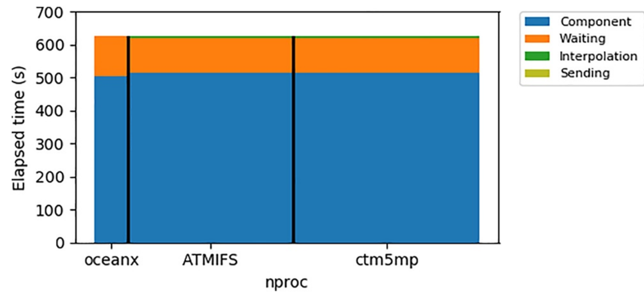

Figure 7. Component and coupling (Waiting, Interpolation, Sending) times for an EC-Earth3-AerChem experiment when all components run at the same speed. The width corresponds to the number of processing elements used by each component.

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 a balanced setup for EC-Earth3-Veg configurations:

- *Memory consumption*: The memory consumption of LPJG is high. On Marenostrum4, it is recommended to use three nodes with 96 GB 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 min 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.

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.

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:

- 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 the memory consumption of these components changes as we reduce the number of cores they use per node.

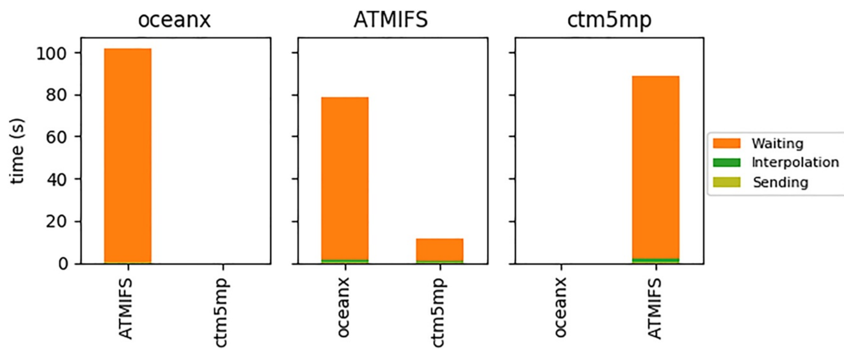

Figure 8. Waiting time between components for an EC-Earth3-AerChem experiment when all components run at the same SYPD. NEMO is always waiting for IFS, IFS is mostly waiting for NEMO and TM5 only waits for IFS.

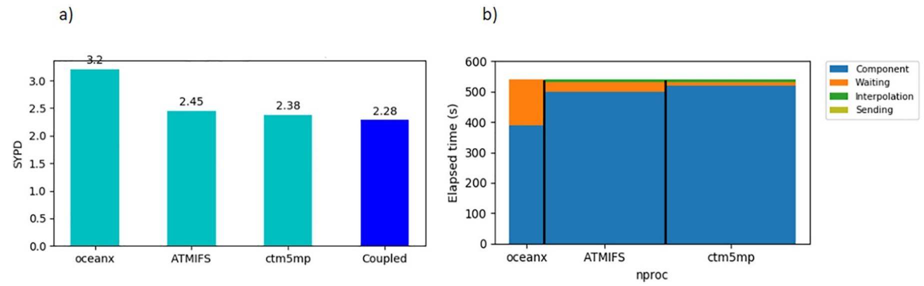

Figure 9. Results using a balanced EC-Earth3-AerChem resource configuration. (a) IFS, NEMO, TM5, and coupled SYPD. (b) Component and coupling (Waiting, Interpolation, Sending) times per component. The width corresponds to the number of processing elements used by each component.

- The communication overhead: If the component shows an overhead due to the extra communication needed between different nodes after scattering its processes, we have to ensure that this loss in efficiency would not be big enough to make the explicit affinity solution unworthy.

In both cases, IFS and NEMO do benefit from reducing the number of cores per node they use and the communication overhead is negligible. According to the results obtained, the memory consumption of IFS is higher and we concluded that it is better to make NEMO and LPJG components share resources. The improved setup uses 336 processes for IFS, 380 for NEMO and 40 for LPJG. A total of eight nodes are used by NEMO and LPJG at the same time, the first running on 43 cores and the latter on the remaining 5. The results are shown in Table 3. Although the coupled SYPD achieved is 5% (13.2/13.9) lower, the number of resources needed has decreased by 40% (768/1,104) and the CHSY is now 28% (1,318/1,824) better.

4.4. EC-Earth3-CC

The last of the configurations to evaluate consists again of IFS, NEMO, and LPJG but it also adds a reduced version of TM5 to simulate the atmospheric Carbon cycle (TM5_CO2). Again, with LPJG we can not use the same SYPD strategy. Instead, we will again show how spreading the physical allocation of its processes is the best approach to minimize the performance loss by this component and how to balance an experiment with four coupled components. TM5_CO2 does not scale very well, but after doing the scalability analysis we found that it is much faster than the full TM5 execution and that instead of using 45 processes (as the default resource configuration suggested), this component is faster when using only eight processes, achieving almost 9 SYPD. Given that TM5 limits the execution speed, we have reduced IFS and NEMO processes to 256 and 192 respectively, so that IFS and TM5_CO2 run at the same speed while NEMO is a bit faster. Note that this is also the maximum number of resources we can give to IFS due to the constraints when running with TM5 described in Section 4.2 As discussed in Section 4.3, we have chosen to spread LPJG processes so that we can reduce the number of resources needed from 144 processes (three full nodes) to only 20. Note that we have reduced a bit more the number of processes used for LPJG even though this increases the initialization phase time. In this case, however,

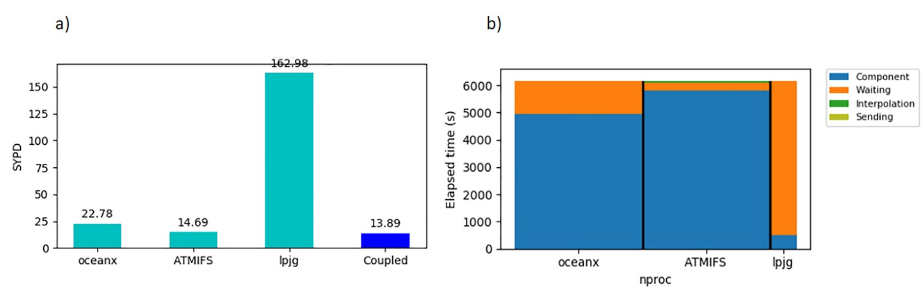

Figure 10. Results using the default EC-Earth3-Veg resource configuration. (a) NEMO, IFS, LPJG, and coupled SYPD. (b) Component and coupling (Waiting, Interpolation, Sending) times per component. The width corresponds to the number of processing elements used by each component.

Table 3
Performance Results of the Original and Balanced Resource Setup for EC-Earth-Veg Experiments

	Original	Balanced
SYPD	13.9	13.2
CHSY	1,824	1,318
Cpl cost (%)	28.7	21
PEs	1,104	768

the number of PEs that will remain IDLE during that time is less than in the EC-Earth-Veg case as we are using fewer cores for IFS and NEMO (due to the TM5_CO2 being slower and limiting the maximum of IFS cores). The results obtained with this new setup are summarized in Table 4. The Coupling cost has been reduced by half, the CHSY has improved by 32% (we use less PEs) and the coupled SYPD is 9% better.

5. Future Work and Conclusions

Achieving the best performance of coupled ESMs is impossible without studying the scalability properties of their constituents and how they are linked during the simulation. Without the right tools and metrics needed to understand the behavior of these complex applications and a well-grounded methodology, we perform Earth System simulations without using the HPC resources effectively due to load-balance issues.

This paper presents the required performance metrics and how to interpret them in order to balance different Coupled Model Intercomparison Project Phase 6 (CMIP6) configurations of EC-Earth3 that couple up to four different components. Furthermore, we have introduced a new metric (Component_cpl_cost) which helps to identify which of the multiple coupled components is the bottleneck of the ESM execution. During our analysis, we have shown that intuitive approaches like running all the constituents at the same speed may lead to suboptimal configurations, we have encountered components that barely scale and limit the speed of the whole coupled model, and components that need extra resources due to their memory requirements. Despite the timestep irregularities that make a perfect load balance impossible for EC-Earth, we were able to achieve configurations with a coupling cost below 10% for experiments using up to four components. In addition, our approach resulted in improved setups in terms of energy and time compared to those previously used at the Barcelona Supercomputing Center. We believe that other ESMs could reduce the coupling cost even further by following our new methodology. Furthermore, we hope to include in new studies the comparisons between our improved resource setups and the ones collected by the community during the CMIP6 runs, including longer simulations and running on different HPC platforms.

In the future, we are expecting ESMs to grow in complexity and in the number of constituents that they will include. Performing these load-balance studies will be key to make the best possible usage of the current and new HPC platforms that are to come. However, the work of manually finding the best resource setup for all the possible configurations of an ESM is very time-consuming and not affordable for many of the teams whose main focus is on the Earth's science, as any change in the model (e.g., components used, grid resolution, output intensity, compilation flags, coupling configuration, etc.) may require to repeat the analysis and tweak the resources used for each particular case. Therefore, to ensure that the optimal solution is found we believe that it would be essential to create a tool that can automatically balance any ESM, finding the appropriate number of PEs to use for any number of coupled components depending on the particular needs of the scientists and bearing in mind

Table 4
Performance Results of the Original and Balanced Resource Setup for EC-Earth-CC Experiments

	Original	Balanced
SYPD	7.1	7.73
CHSY	2,104	1,428
Cpl cost (%)	33.7	15.1
PEs	621	476

the existing HPC platform constraints (e.g., time vs. energy solutions, queue limitations on the wall-clock or on the maximum number of cores, etc.).

Data Availability Statement

The scalability plots and data for stand-alone executions of EC-Earth3 components can be found in the following GitLab repository: https://earth.bsc.es/gitlab/spalomas/ec-earth3-scalability-analysis.
The sources for EC-Earth3 ESM can be found on the main web page: https://ec-earth.org/.
Bear in mind that due to IFS code license of ECMWF, the development portal (SVN repository) can only be accessed by the EC-Earth consortium.
Finally, the OASIS-MCT3 coupler sources can be found on their main GitHub page: https://gitlab.com/cerfacs/oasis3-mct/-/tree/OASIS3-MCT_3.1.

Acknowledgments
This project has been a lot of work, but we couldn't have done it without the support and guidance from some very important people. We want to thank Dr María Gonçalves and Mr Miguel Castrillo for all their help with this project; they provided us with resources as well as essential information that was needed to complete our task successfully. We also want to thank all of the people within the Computational Earth Science group at the BSC for working alongside us on this project and providing the crucial tools to achieve our goals. The research leading to these results has received co-funding from the National Research Agency through OEMES (PID2020-116324RA-I00).

References

Acosta, M., Palomas, S., & Paronuzzi, S. (2021). IS-ENES3 D4.3—CPMIP performance metrics and community advice. https://doi.org/10.5281/ZENODO.6394049

Acosta, M., Yepes-Arbós, X., Valcke, S., Maisonnave, E., Serradell, K., Mula-Valls, O., & Doblas-Reyes, F. (2016). Performance analysis of EC-Earth 3.2: Coupling. Retrieved from https://earth.bsc.es/wiki/lib/exe/fetch.php?media=library:external:technical_memoranda:bsc-ces-2016-006-coupling_ec-earth.pdf

Balaji, V. (2015). Climate computing: The state of play. *Computing in Science & Engineering*, *17*(6), 9–13. https://doi.org/10.1109/MCSE.2015.109

Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., et al. (2017). CPMIP: Measurements of real computational performance of Earth system models in CMIP6. *Geoscientific Model Development*, *10*(1), 19–34. https://doi.org/10.5194/gmd-10-19-2017

Donners, J., Basu, C., Mckinstry, A., Asif, M., Porter, A., Maisonnave, E., et al. (2012). Performance analysis of EC-Earth 3.1. PRACE whitepaper.

Döscher, R., Acosta, M., Alessandri, A., Anthoni, P., Arsouze, T., Bergman, T., et al. (2022). The EC-Earth3 Earth system model for the Coupled Model Intercomparison Project 6. *Geoscientific Model Development*, *15*(7), 2973–3020. https://doi.org/10.5194/gmd-15-2973-2022

Hanke, M., Redler, R., Holfeld, T., & Yastremsky, M. (2016). YAC 1.2.0: New aspects for coupling software in Earth system modelling. *Geoscientific Model Development*, *9*(8), 2755–2769. https://doi.org/10.5194/gmd-9-2755-2016

Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., et al. (2014). C-Coupler1: A Chinese community coupler for Earth system modeling. *Geoscientific Model Development*, *7*(5), 2281–2302. https://doi.org/10.5194/gmd-7-2281-2014

Maisonnave, E., Coquart, L., & Piacentini, A. (2020). A better diagnostic of the load imbalance in oasis based coupled systems (Tech. Rep.), TR/CMGC/20/176, CECI, UMR CERFACS/CNRS No5318.

Ticco, S., Acosta, M., Castrillo, M., Tintó, O., & Serradell, K. (2020). Keeping computational performance analysis simple: An evaluation of the nemo bench test (Tech. Rep.), Partnership for Advanced Computing in Europe, PRACE white paper.

Valcke, S. (2013). The OASIS3 coupler: A European climate modelling community software. *Geoscientific Model Development*, *6*(2), 373–388. https://doi.org/10.5194/gmd-6-373-2013

Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., et al. (2012). Coupling technologies for Earth system modelling. *Geoscientific Model Development*, *5*(6), 1589–1596. https://doi.org/10.5194/gmd-5-1589-2012

Will, A., Akhtar, N., Brauch, J., Breil, M., Davin, E., Ho-Hagemann, H. T. M., et al. (2017). The COSMO-CLM 4.8 regional climate model coupled to regional ocean, land surface and global Earth system models using OASIS3-MCT: Description and performance. *Geoscientific Model Development*, *10*(4), 1549–1586. https://doi.org/10.5194/gmd-10-1549-2017