This is an accepted manuscript, the final version can be found at:https://dl.acm.org/doi/abs/10.1145/3624062.3624283
https://doi.org/10.1145/3624062.3624283

End-to-end Workflows for Climate Science: Integrating HPC Simulations, Big Data Processing and Machine Learning

Donatello Elia*
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Lecce, Italy

Sonia Scardigno
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Lecce, Italy

Jorge Ejarque
Barcelona Supercomputing Center
Barcelona, Spain

Alessandro D'Anca
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Lecce, Italy

Gabriele Accarino
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Lecce, Italy

Enrico Scoccimarro
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Bologna, Italy

Davide Donno
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Lecce, Italy

Daniele Peano
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici
Bologna, Italy

Francesco Immorlano
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici & Department of Innovation
Engineering, University of Salento
Lecce, Italy

Giovanni Aloisio
Fondazione Centro
Euro-Mediterraneo sui Cambiamenti
Climatici & Department of Innovation
Engineering, University of Salento
Lecce, Italy

ABSTRACT

Current scientific workflow systems do not typically integrate simulation-centric and data-centric aspects due to their very different software/infrastructure requirements. A transparent integration of such components into a single end-to-end workflow would lead to a more efficient and automated way for generating insights from large simulation data. This work presents a complex case study related to extreme events analysis of future climate data that integrates in the same workflow numerical simulations, Big Data analytics and Machine Learning models. The case study is being implemented in the context of the eFlows4HPC project using the project's software stack for deployment and orchestration of the workflow. The solution implemented in the project has shown to simplify the development and execution of end-to-end climate workflows with heterogeneous software requirements. Moreover, such an approach can, in the long term, increase the reuse of workflows by scientists and their portability over different HPC infrastructures.

CCS CONCEPTS

• **Software and its engineering** → **Data flow architectures**; • **Applied computing** → *Environmental sciences*; • **Computing methodologies** → *Distributed algorithms*; Machine learning.

KEYWORDS

Scientific workflow management, high performance data analytics, data-driven models, extreme climate events

1 INTRODUCTION

The advent of novel computing solutions, jointly with the exponential growth of data, have caused in recent years a radical change in the scientific discovery process in several domains, including climate sciences [6]. This led to the definition of a new *data-centric scientific discovery paradigm*, alongside the well-established *simulation-centric* one, where data became a central component of science [30]. As the complexity of the computing infrastructures and the size of datasets to be handled increase, scientific research has become reliant on technologies able to efficiently handle such large scales [28]. The high availability of data also pushed forward the use of Machine Learning (ML) techniques in scientific research [32], which further increased the plethora of tools available to scientists and, in turn, the complexity of the workflows.

Current scientific workflows, however, do not typically integrate simulation-centric and data-centric aspects of research due to their very different, sometimes orthogonal, infrastructure requirements [2]. End-to-end workflow solutions, capable of handling the whole workflow from numerical model simulation to data processing and visualization, would represent very valuable solutions for speeding

*Corresponding author: donatello.elia@cmcc.it

up the research process and improving scientists' productivity [3]. Such solutions would also allow supporting execution on software stacks with different computing paradigms (i.e., High Performance Computing (HPC) and Cloud). The integration of computing and data-intensive components is, thus, seen as a critical to fully support future scientific discovery [39].

In particular, in the climate science field, which is the main domain targeted in this work, Earth System Models (ESMs) simulations represent one of the most challenging HPC use cases. Indeed, ESMs incur in very high computational cost, intensive Input/Output patterns, very large data volumes produced, which, in turn, drive the necessity for data-intensive post-processing to extract relevant information and knowledge. ESMs simulations rely on large compute-intensive infrastructure, while post-processing, analytics and ML require more data-centric ones: a transparent and seamless integration of components with such heterogeneous requirements into a single end-to-end workflow is challenging but would lead to a more efficient and automated way for generating climate data products.

This work presents a complex workflow related to extreme events analysis of future climate projections from a high-resolution ESM. It introduces the solutions implemented for handling such a challenging case study that integrates into a single end-to-end workflow a numerical simulation, Big Data analytics and ML models. Such effort is being carried out in the context of the *EuroHPC eFlows4HPC project*[1], whose goal is to design and implement an European platform that supports the development of workflows integrating HPC processes, data analytics and artificial intelligence [18]. It will deliver the eFlows4HPC software stack to enable the integration of components with different deployment requirements, spanning from HPC to Cloud computing, and develop the concept of the *HPC Workflows as a Service* (HPCWaaS) to facilitate the use and reuse of workflows across HPC infrastructures.

More in details, the main contribution of this paper can be summarized as:

- Review the challenges, requirements and opportunities for end-to-end climate science workflows;
- Design and implement an end-to-end workflow concerning a complex case study for climate extremes processing on ESM data. This workflow integrates multiple components with heterogeneous software requirements;
- Present the advantages from the application of the eFlows4HPC software stack and the HPCWaaS concept for a end-to-end workflow in climate science.

The rest of this paper is organized as follows: Section 2 describes some background work related to scientific workflows management; Section 3 provides a characterization of challenges and opportunities for end-to-end workflows in climate sciences; Section 4 introduces the software solutions developed in the eFlows4HPC project; Section 5 presents the case study for climate extreme events from ESM simulation data and its main building blocks; Section 6 describes the workflow implementation and some insights from its execution; finally, Section 7 draws the main conclusions from this study and presents some aspects worth of future investigation.

[1] eFlows4HPC website: https://eflows4hpc.eu/

2 BACKGROUND WORK

Workflows allow scientists to code the tasks of a scientific procedure into a precise and descriptive document that can be repeated and executed in a systematic and automated manner. The workflow definition can include a very large set of tasks, representing diverse types of actions, such as scripts, execution of binary programs, invocation of web services, data movement, etc. Dependencies among tasks can be defined to control the data flow and the order in which tasks are executed [26]. Additionally, scientific workflows can promote Open Science practices since the document can easily become compliant with the FAIR principles (Findable, Accessible, Interoperable, Reusable [44]).

The Workflow Management System (WMS) and its runtime represent the components devoted to interpreting the workflow description and managing its whole lifecycle according to the dependencies specification among the tasks. To support compute/data-intensive workloads, the WMS can schedule the execution of the workflow tasks in parallel exploiting different techniques (e.g., independent tasks or sub-workflows are executed concurrently) [33].

Most systems typically support workflows defined as Directed Acyclic Graphs (DAGs) of tasks, where no cyclic dependencies among tasks are allowed, although some solutions also support cyclic interaction among them [23]. WMSs also differ in terms of interfaces provided for workflow development and submission, including graphical user interfaces (GUIs), textual interfaces and more programmatic ones (APIs) [3]. Moreover, workflow systems provide capabilities such as resilience and fault detection, optimized scheduling/execution of tasks, provenance tracking, workflow validation and monitoring, which are some of the key functionalities for supporting large-scale workflows [14].

Different WMSs are currently being employed in various scientific disciplines, such as life sciences, solid Earth sciences, physics, astrophysics and environmental sciences [27, 9, 8, 13, 3]. For example, in the context of the ESM community some of the most used solutions include Cylc [37], ecFlow[2] and AutoSubmit[3]. Such solutions have different implementations and capabilities, but all share the same main goal: supporting operations of large ESM simulations, mainly focusing on the execution of numerical models on HPC facilities [35].

Comprehensive integration of HPC-based simulations, data analytics and ML processing with the current workflow solutions is not trivial and novel solutions are needed [4]. In particular, some of the challenges that need to be addressed in large scientific (and climate) workflows involve: (i) simplifying the development of complex workflows taking into account all the components of an end-to-end ESM workflow, including simulations, Big Data and ML components, (ii) supporting the execution on heterogeneous computing environments (e.g., HPC and Cloud), and (iii) enabling flexibility and dynamicity in the workflow execution.

[2] ecFlow documentation: https://ecflow.readthedocs.io/en/latest/index.html
[3] AutoSubmit documentation: https://www.bsc.es/research-and-development/software-and-apps/software-list/autosubmit/documentation

3 CHALLENGES, REQUIREMENTS AND OPPORTUNITIES IN ESM WORKFLOWS

Typical ESM workflows can include different steps such as: input data preparation, multiple numerical simulation runs, output data post-processing, data analytics, and visualization [1]. Usually, ESM simulation workflows focus on the model run and the storage of output data, while data analytics, interactive analysis and visualization are performed by scientists in a second stage using custom tool and scripts. In some cases, a part of the analysis is already performed online during model simulations with the goal of pre-computing some relevant statistics or simple indicators useful for validating the results (e.g., diagnostics) [45]. However, in-depth and complex analyses of the simulation outputs, requiring for example access to multiple variables and longer time ranges or including advanced algorithms, are typically executed outside the ESM simulation workflow.

It is important to remark that ESM simulations and data analytics exhibit different needs in terms of software infrastructure, execution time frame and I/O requirements.

ESMs are based on numerical modelling requiring large HPC infrastructure for running the simulation in parallel (i.e., using MPI and OpenMP) in a reasonable amount of time. The time scale for execution of such simulations may vary from a few hours up to several weeks according to the spatial resolution, the complexity of the physical processes modeled and the number of simulation runs in the ensemble (group of runs of the same ESM with different initial conditions) [15]. In terms of I/O, emphasis is usually placed on maximizing writing performance of the model output.

Analysis of ESM data employs tools for statistical analytics, data mining, interactive exploration, visualization, and more recently High Performance Data Analytics (HPDA) and ML techniques for extracting knowledge from large volumes of climate data. Such applications are typically embarrassingly parallel and mainly based on data-driven approaches where huge data volumes need to be efficiently accessed and processed. Most of the software for data analytics has been developed for being executed in highly distributed storage infrastructures, typical of Cloud environments.

These different requirements have hindered the integration and co-existence of compute-driven and data-driven components into single applications. A more integrated approach would allow supporting next-generation ESMs and improve the workflow in terms of execution and energy consumption. In particular, the integration of the compute and data-driven stages into a single end-to-end ESM workflow could bring several advantages:

- Automating the whole workflow from ESM execution to data analysis can speed up the generation of climate data products improving scientist productivity. Additionally, the workflow management systems can control the status of all the tasks, thus supporting error management in a uniform manner;
- End-to-end workflow systems can provide a standardized way to develop workflows integrating the different ESM simulation runs along with data analytics and ML components. In turn, this can simplify the overall workflow implementation process a support reusability in the climate community;
- A single WMS can take into account the requirements of the different tasks and transparently deploy their execution on the most suitable computing and software infrastructure, as well as enabling flexible and efficient scheduling of the tasks composing the workflow;
- Seamless integration of ESM simulation and data processing into a single workflow can allow for better optimization in terms of data movement and access. Data could be, in fact, kept in memory and moved to other nodes as the workflow progresses through the various tasks.

4 THE EFLOWS4HPC SOFTWARE STACK

4.1 The HPCWaaS approach

eFlows4HPC aims to provide a *software stack* that facilitates the development, deployment and execution of workflows combining HPC simulation and modeling with artificial intelligence and data analytics. With this goal, the project is also developing the concept of the *HPC Workflows as a Service* (HPCWaaS), which leverages the software stack components to provide a mechanism to facilitate the use and reuse of workflows in HPC infrastructures.

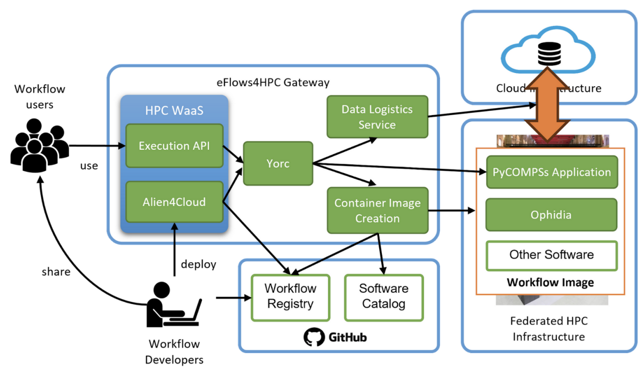

Figure 1: Overview of the eFlows4HPC HPC Workflows as a service (HPCWaaS) methodology.

Figure 1 describes how the proposed HPCWaaS methodology works. It provides two interfaces: the one used by workflow developers for developing the workflows descriptions and deploying them in the HPC infrastructures; and another one used by the final user for executing the workflows. The development interface is provided by *Alien4Cloud*[4], a GUI to describe the topology of components involved in the workflow deployment and execution in an extended TOSCA format [42]. The goal is that, given this topology and a HPC cluster location, it can be used by *Yorc*[5] (a TOSCA orchestrator) for deploying the software and data required to execute the workflow. The Software deployment is done by means of the *Container Image Creation service* which automates the creation of the container images for workflows, including the code as well as all the required software compiled for the target HPC platform [16]. On the other hand, the management of the required data is done by the *Data*

[4]Alien4Cloud documentation: https://alien4cloud.github.io/#/documentation/3.5.0/getting_started/new_getting_started.html
[5]Yorc Github: https://github.com/ystia/yorc

Logistics Service[6] which executes the required data pipelines either at deployment or execution time. Once the workflow is deployed, it is published to the *HPCWaaS Execution API* which allows final users to run the deployed workflow as a simple REST invocation. As a result of such invocation, the workflow execution is triggered in the HPC system, with PyCOMPSs managing its run inside the computing infrastructure.

4.2 Components

The eFlows4HPC software stack comprises several software components[7]; this section focuses on the key components exploited in the presented case study for extreme events analysis on ESM projection data.

4.2.1 PyCOMPSs. PyCOMPSs [41] is a task-based programming model that focuses on making the development of parallel applications easier for distributed computing. Its syntax uses annotations to identify the methods that will become tasks at execution time and a small API for synchronization. To declare a task, a Python function is annotated with the *@task* decorator which includes the directionality clauses for the function parameters (i.e., *IN* indicates data used by the task, *OUT* indicates data created in the task, *IN-OUT* indicates data modified in the task). The annotated Python script is interpreted by the COMPSs Runtime [5] which converts the script to a parallel workflow. Every time an annotated method (task) is invoked from the script, the COMPSs runtime creates a node in a task-graph and looks for data dependencies with previous existing tasks according to the declared data directionality. The runtime is then able to exploit the potential parallelism of the task graph by scheduling those tasks that do not have data dependencies between them. The runtime is also able to execute tasks in an asynchronous fashion, starting new tasks once their predecessors end. The COMPSs runtime handles all data transfers automatically, by moving data on-demand between the computing nodes of the infrastructure. However, when a task result is needed in the main program, data need to be sent (synchronized) to the computing node where the main program is executed once the task ends.

PyCOMPSs tasks can also support heterogeneous computing. Developers can define tasks targeting different processors and accelerators (such as numpy or cupy) using the *@constraint* decorator, or integrate with other programming paradigms including other decorators (such as *@mpi*). With regards to fault tolerance, a mechanism at task level is provided, where the programmer can indicate in a decorator the behavior to implement in case of task failure (i.e., ignore the failure of the task and continue, stop the whole workflow, etc.) [17]. A checkpointing mechanism at task level has also been implemented, which enables to recover a failed execution from the last checkpointed task [43].

PyCOMPSs applications are platform agnostic so they can be executed in different types of infrastructures: clouds, large clusters (or supercomputers) and container-managed clusters [38]. They are deployed following the master-worker paradigm, where the master runs the main python script and the workers run the tasks in the different computing nodes.

[6]Data Logistics Service: https://gitlab.jsc.fz-juelich.de/eflows4hpc-wp2/data-logistics-service
[7]eFlows4HPC software stack: https://eflows4hpc.readthedocs.io/en/latest/

4.2.2 PyOphidia. PyOphidia represents a Python module for large-scale data analytics on scientific multi-dimensional data [21]. It provides the Python bindings for the Ophidia framework, an open source software solution for scientific data analytics developed by CMCC [24]. The Ophidia framework joins parallel computing paradigms and big data management approaches for supporting scientific HPDA. It implements an array-based storage mode, leveraging the datacube abstractions from data warehouse systems, and a hierarchical storage organization for partitioning and distributing large multi-dimensional scientific datasets over multiple nodes. It provides, among the others, features for time series processing, data reduction, subsetting, statistical analysis, data intercomparison, NetCDF I/O, datacube manipulation, and metadata management. Although Ophidia core features are related to analysis of ESM data, it has successfully been used also in other scientific contexts (e.g., astronomy, seismology, biodiversity [25]).

From an architectural perspective, the Ophidia framework follows a client-server approach, where the client-side components (e.g., PyOphidia) dispatch the execution of the data processing tasks on the server-side, deployed near the HPC or Cloud infrastructure. The server-side part of the framework includes a main front-end service, the Ophidia Server, and the computing components, the runtime and the in-memory I/O Servers. The front-end server and the computing components are usually deployed on different nodes of the infrastructure. Furthermore, the number of Ophidia computing components can be scaled up, also dynamically, over multiple nodes of the infrastructure to address more intensive data analytics workloads [19]. To this end, the framework supports integration with different HPC scheduling systems.

4.2.3 CMCC-CM3 Earth System Model. Although the ESM is not properly part of the eFlows4HPC software stack, it is one of the key components of the extreme events analysis workflow and it can be orchestrated by the PyCOMPSs runtime. The ESM considered in the case study is the latest version of the CMCC global coupled model: the *CMCC-CM3 climate model*. It is based on the *Community Earth System Model (CESM)* project[8] operated at the National Centre for Atmospheric Research (NCAR) in the United States, and used to run Coupled Model Intercomparison Project, phase 6 (CMIP6) [22] simulations following both simulation scenarios and HighResMIP protocols [29]. The CMCC-CM3 oceanic component is based on the Nucleus for European Modelling of the Ocean (NEMO) model, version 4.0, while the atmospheric component is the Community Atmosphere Model version 6 (CAM6). The adopted spatial resolution is ¼degree, corresponding to 25 km x 25 km of grid spacing in both atmosphere and ocean components. Every few minutes the heat, momentum and mass fluxes are sent from the atmosphere to the ocean and the sea surface temperature, the sea ice cover and the surface velocities are sent from the ocean to the atmosphere. In this way the fully coupled system is able to evolve in time, without any external support except for the greenhouse gases concentrations, that are provided year by year through I/O, corresponding to historical concentrations and/or future plausible projections. CMCC-CM3 is the model in preparation for the next CMIP7 effort[9].

[8]CESM: http://www.cesm.ucar.edu
[9]CMIP7: https://wcrp-cmip.org/cmip7/

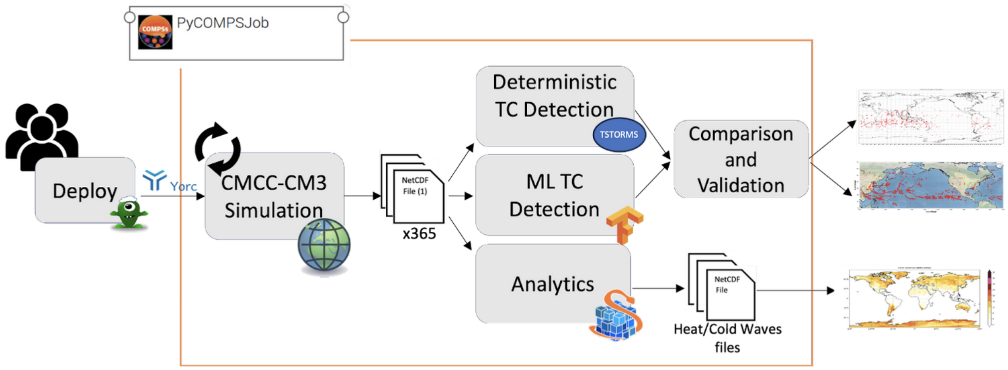

Figure 2: High-level view of the workflow for the climate extreme events case study. Deployment and submission of the workflow is handled by Alien4Cloud, while the actual workflow execution is orchestrated by PyCOMPSs.

5 A CASE STUDY FOR CLIMATE EXTREME EVENTS ON ESM FUTURE DATA

5.1 General workflow

The selected case study targets the analysis of climate extreme events (such as heat waves and tropical cyclones) from ESM projection outcomes. Understanding how climate change affects such events is very important for policy makers to develop mitigation strategies and monitoring systems, since extreme events can have severe impacts on the economy and people's life [7, 12]. The Intergovernmental Panel on Climate Change (IPCC) has dedicated a chapter of the *sixth Assessment Report* (AR6) on the changes in weather and climate extremes highlighting, in most cases, an increase in their intensities and frequencies [40].

From a technical standpoint, due to their rare nature, analysis of extreme events requires processing of huge amounts of data. To this end, HPDA can help in preparing the data for the analysis and running descriptive analytics in parallel, while novel ML-based approaches can improve, in terms of efficiency and accuracy, the prediction and localization of such extreme events [34].

Combining into a single workflow the numerical ESM execution (i.e., CMCC-CM3 model) together with the extraction of knowledge based on HPDA and ML techniques is quite challenging, due to the types of diverse software that need to be managed during the workflow lifecycle. Nevertheless, their integration into a single end-to-end workflow can help in reducing the overall execution time as different tasks of the workflow can be executed concurrently. In this way, as the model starts to produce its output, the data processing required for statistical analysis and ML model inference can seamlessly be executed on different HPC nodes.

PyCOMPSs was exploited for implementing the case study workflow as a Python application. Several tasks were defined to handle the different stages of the workflow and integrate the required software: the CMCC-CM3 model, PyOphidia for climate data analytics, Keras [11] and Tensorflow [36] for the ML models, as well as a set of additional Python dependencies.

The resulting workflow description, stored in the eFlows4HPC workflow registry, is accessed via the HPCWaaS interface and executed on a selected HPC infrastructure. Figure 2 shows a high-level view of the case study workflow. More in details, the whole process consists of the following steps:

(1) Starting from the workflow description and the yaml TOSCA file describing the application architecture in terms of the software and data requirements, the Alien4Cloud interface is used for defying the application parameters and the HPC endpoint, deploying the environment for the workflow (through the Yorc orchestrator, see Section 4.1) and starting its execution;
(2) The PyCOMPSs application execution is triggered on the HPC infrastructure and the different tasks within the application are orchestrated according to their dependencies;
(3) The first task being executed consists of the CMCC-CM3 model simulation. This task runs iteratively for producing the output data (one NetCDF file for each day of simulation) until the simulation run is completed;
(4) Concurrently with the model run, as soon as full year of NetCDF files is available, the data analytics and ML tasks are executed on the new daily variables produced by the simulation for performing:
    (a) The computation of the Heat/Cold Waves indicators;
    (b) The localization of Tropical Cyclones via i) pre-trained ML model(s) and ii) a deterministic algorithm for Tropical Cyclones tracking;

    more details on these two steps are reported in the following subsections;
(5) As the processing in step 4 progresses, the output of the analysis is then validated and stored on disk as NetCDF files;
(6) Once the model simulation and the related processing is completed, maps can be produced starting from the results stored on disk (or retained in memory). It must be noted that intermediate maps with the results of the extreme event analysis for single years of data can be already produced during step 5, while the final plots/maps are produced from the whole set of final validated results.
(7) As the PyCOMPSs application is completed, the workflow is undeployed and the status is returned to the Alien4Cloud interface.

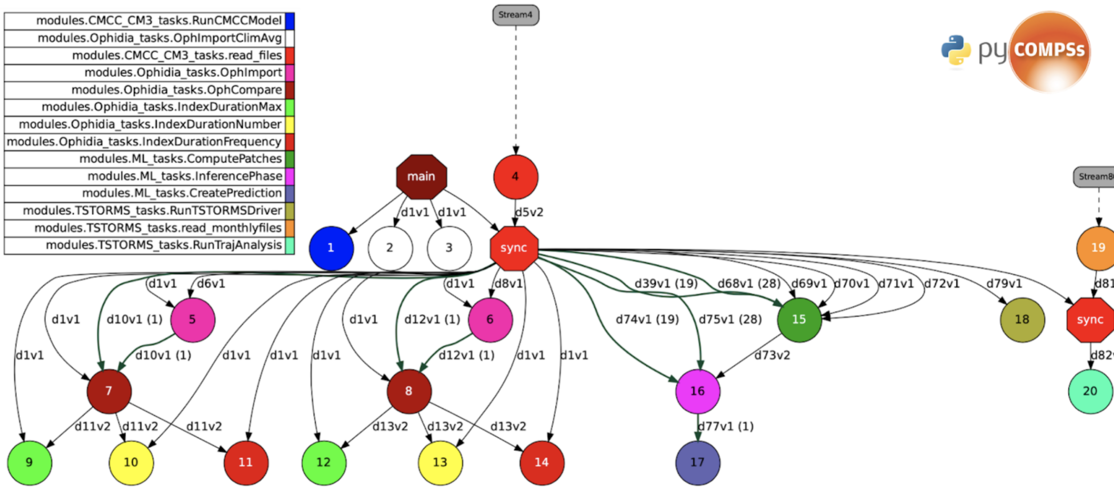

Figure 3: Graph representation of the climate extreme events workflow produced at run time by PyCOMPSs. Each circle represents a workflow task. Different color represent the different function/method defined in the Python code.

The overall workflow graph structure produced by PyCOMPSs at run time is shown in Figure 3. Each circle in the figure represents a different task implemented in the Python workflow and orchestrated by PyCOMPSs. For the sake of readability, the graph shows a simplified version of the workflow running for a single year of simulation data; in case of multiple years, the number of tasks (i.e., circles) would be repeated with the exception of the first four ones related to ESM run and preliminary data loading (from #1 to #4). As it can be seen, even in this simplified version, the workflow structure is quite complex and multiple dependencies are defined among the different tasks with the aim of efficiently reusing data between the different steps of the workflow. The following subsections provide additional details on main building blocks of the climate extreme events workflow.

5.2 ESM execution

The CMCC-CM3 simulation block represents the first step in the general graph (blue circle with #1 in Figure 3). It produces daily NetCDF files of size 271 MB with dimensions of 768 (latitudes) x 1152 (longitudes) x 4 (6-hourly timesteps) including around 20 single precision floating point variables (e.g., precipitation rate, sea level pressure, temperature, wind speed, etc.). The execution of the subsequent steps of the workflow starts when the files for a whole year are available in the directory (i.e., nearly 100 GB). In the meantime, the simulation will continue running for producing the data for the other years. Supporting concurrent execution of the ESM simulation and post-processing is key since the projection time span of the climate simulation can consist of multiple tens of years (e.g., 30-35 years) and require several days (up to a few months) to complete, according to the HPC infrastructure used. For enabling the concurrent execution of tasks, a streaming interface available in PyCOMPSs has been leveraged to monitor the file production progress and detect when a (full) new year of data is available (red circle with #4).

5.3 Heat/Cold Waves indices computation

As soon as a full year of the CMCC-CM3 output is available, pipelines of data analytics operators are executed for the computation of multiple extreme climate events indices for heat waves and cold spells (or cold waves). A heat wave is a period of unusually hot weather that typically lasts six or more days. To be considered a heat wave, the maximum temperature must be 5 °C higher than the historical averages (e.g., computed over a 20-year period) for a given area; conversely for a cold wave the minimum temperature must be 5 °C lower than the historical averages [31]. In particular the indices computed are maps with: (i) the longest heat/cold wave duration per year (green tasks with #9 and #12 in Figure 3), (ii) the number of heat/cold waves per year (yellow tasks with #10 and #13) and (ii) the frequency of yearly heat/cold waves (red tasks with #11 and #14).

As the amount of data to handle is large, HPDA operators from the Ophidia framework are used for processing and aggregating the dataset in parallel. Moreover, since Ophidia can store the datasets in memory between different operators' execution [20], the baseline values with the long-term historical averages can be loaded only once and used throughout the workflows for the computation of the indices, reducing the number of read operations from storage. The complete sub-workflow includes the steps for loading and post-processing the model data (in particular the minimum and maximum temperature variables), for computing the indices and for producing the output as NetCDF files and maps. Figure 4 shows an example of an Heat Wave indicator resulting from the sub-workflow execution.

5.4 Tropical Cyclones detection and tracking

Tropical Cyclones (TCs) are complex phenomena driven by a combination of atmospheric and oceanic processes. ESMs allow simulating these physical interactions, providing valuable insights into their genesis, intensification and tracks. However, detecting such extremes in large climate datasets remains a challenging task, mainly

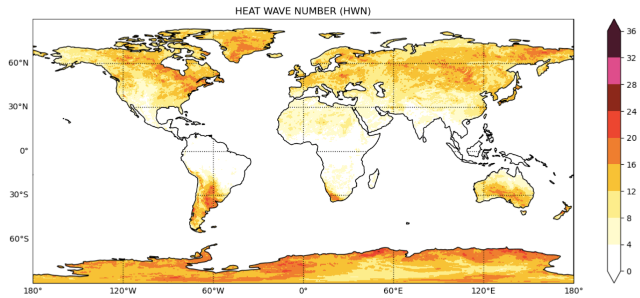

Figure 4: Heat Wave Number indicator computed on a year of simulation data produced by the CMCC-CM3 model. For each point of the map the number of heat waves occurred during the year is displayed.

due to the large data volumes to be handled and the limitations of traditional tracking schemes [10]. Machine learning approaches can be used for extracting significant spatial features related to the presence of TCs in gridded climate data. Integrating ESM simulations and a ML-enabled TC localization algorithm in a comprehensive end-to-end workflow would support the assessment of TCs impacts, as well as the indication of their frequency, intensity, and distribution on simulation data.

The ML-enabled TC localization approach implemented in this case study allows identifying the presence of TC given a set of input climate variables simulated by ESM (i.e., temperature, sea pressure level, wind speed, vorticity) and localizing its center (or "eye") in terms of its geographical coordinates (i.e., latitude and longitude). A convolutional neural network (CNN) previously trained on historical data can be used for localizing the TCs centers. The sub-workflow involves different tasks for: (i) post-processing of model simulations (i.e., regridding the CMCC-CM3 file, tiling of data into non-overlapping patches, feature scaling, etc.), (ii) inference through the pre-trained CNNs and (iii) geo-referencing predicted TC center coordinates onto a global map, providing visualization utilities (respectively represented by the green #15, magenta #16 and purple #17 circles in Figure 3). Moreover, the workflow for climate extreme events can execute deterministic TC tracking schemes to further validate the results.

6 TEST BED IMPLEMENTATION AND ADDED VALUE OF THE EFLOWS4HPC SOLUTION

The case study was executed in a geographically distributed testbed with the HPCWaaS interface running on a service node at the Barcelona Supercomputing Center (BSC) at Barcelona, Spain, while the entire workflow was executed on the Zeus supercomputer at the Euro-Mediterranean Center on Climate Change (CMCC) at Lecce, Italy. Zeus is one of the HPC systems available at the CMCC supercomputing center[10]. It delivers 1.2 PetaFlops of peak performance and is composed of 348 nodes with a total of 12,528 processors and 33.4 TB of main memory. The cluster exploits a GPFS parallel file system and IBM Spectrum LSF as scheduling system.

[10]CMCC SuperComputing Center: https://www.cmcc.it/super-computing-center-scc

The Alien4Cloud service running at BSC was used to deploy the environment at the Zeus CMCC cluster and submit the workflow by remotely interacting with the cluster scheduling system according to the configuration defined in the TOSCA topology. PyCOMPSs orchestrates the execution of the single workflow tasks on Zeus, following the steps reported in Section 5.

In this first setup of the testbed all the components were running on Zeus either bare-metal or by using Python environments. However, through the HPCWaaS solution, containers (e.g., Singularity) with the software required by the workflow, when supported by the systems, can be exploited for running the application.

The implementation of the case study on extreme events analysis from climate simulation proved to be challenging since the different steps of the workflow exhibit different software and execution requirements. Nevertheless, the use of the eFlows4HPC software stack allowed to simplify the development process and the management of the ed-to-end workflow.

From a developer point of view, the different tasks of the workflow can be coded simply as Python functions and annotated with PyCOMPSs decorators. Listing 1 shows an extract of the sub-workflow code for heat/cold waves computation including a couple of tasks for the indices computation (defined in Section 5.3).

```
@task(returns=object)
def IndexDurationMax(client, duration, filename):
    cube.Cube.client = client
    #Maximum lenght of heat/cold waves in a year
    Max = duration.reduce(operation='max',...,
        description="Max Duration cube")
    Max.exportnc2(output_path=OUTPUT_PATH,
        output_name=filename)
    return Max

@task(returns=object)
def IndexDurationNumber(client, duration, filename
    ):
    cube.Cube.client = client
    #Number of heat/cold waves in a year
    Mask = duration.apply(query="oph_predicate('
        OPH_INT','OPH_INT',measure,'x
        ','>0','1','0')", ...)
    Count = Mask.reduce(operation='sum', ...,
        description="Number of durations cube")
    Mask.delete()
    Count.exportnc2(output_path=OUTPUT_PATH,
        output_name=filename)
    return Count
```

Listing 1: Python code snippet from the sub-workflows for heat/cold waves indices computation. @task decorator from PyCOMPSs is used for defining workflow tasks, while PyOphidia is used to perform data reduction operators.

Using Python as the workflow programming language made the case study implementation and the integration of climate processing modules with libraries for Big Data and ML frameworks easier. Through this integration, multiple levels of parallelism can be easily supported, as PyCOMPS can automate concurrent execution of independent tasks on different NetCDF files produced by the simulation, while PyOphidia can run climate analytics in parallel on each set of files. Moreover, in the implemented workflow, tasks related to climate indices computation and TC localization can start as soon as enough data are available from the model and run concurrently with the ESM simulation. The results from the analysis are, thus, available to the scientist while the simulation progresses.

From an end user point of view, once the workflow is coded, it can be easily scheduled and run on the HPC infrastructure through the Alien4Cloud GUI, relieving the user from the burden of setting up the environment. Input arguments can be specified to configure the workflow, while the execution of the tasks for indices computation and TC centers localization can dynamically adapt to the number of files produced by the ESM. In this way, climate scientists can focus more on the results of the simulations and related analysis, rather than handling complex workflows and setting up the software environment.

The HPCWaaS approach from eFlows4HPC can also increase the portability and reusability of the workflows by supporting the execution of workflows on different HPC centers. The workflow can be, in fact, reused by scientists and run on a different HPC systems without too much effort, since the software requirements will be handled by Yorc following the related TOSCA description. Moreover, the HPCWaaS solution can also support the execution of different components of the workflow in a geographically distributed setup. Although the current version of the climate extreme events workflow was completely executed on a single HPC system, we plan to extend the implementation in the future to support a distributed execution of different tasks by leveraging the Data Logistics Service from the eFlows4HPC software stack for data movement. To this extent, the different parts of the workflow could be run on different infrastructures according to their requirements, using, for instance, large HPC systems for the ESM simulation, data-oriented/Cloud systems for Big Data processing and GPU-partitions for the ML-based models.

7 CONCLUSIONS AND FUTURE WORK

This work addresses the challenges related to end-to-end workflows on HPC infrastructures in the climate domain. It presents a challenging case study, developed in the context of the eFlows4HPC project, on climate extreme events analysis and predictions integrating into a single workflow high-resolution ESM simulations and data-driven analyses.

Current solutions in the context of climate science have limitations in fully supporting workflows integrating numerical simulation, Big Data analytics and ML models, mainly due to their different software and execution requirements. The eFlows4HPC project aims at providing a software stack for enabling the integration of components with diverse deployment requirements and supporting the use and reuse of workflows over different HPC infrastructures through the HPCWaaS concept.

The proposed case study was implemented as a Python application exploiting components from the eFlows4HPC software stack together with other well-known and community-based modules, resulting in a simpler development and management of its workflow. Additionally, the HPCWaaS interface allowed the end-users (i.e., the climate scientists) to run the workflow on a HPC system without the need to directly interact with the infrastructure, further improving the overall productivity.

Hence, *software solutions/approaches like those described for the extreme events case study could be exploited in the future to effectively support other end-to-end workflows in climate sciences and to take advantage of Big Data and Machine Learning in operational scenarios*. Such approaches will lead to (i) quicker and more automated generation of insights and indices from the simulation outputs and (ii) a higher reuse and portability of such complex climate sciences workflows on different HPC infrastructures.

Future work will focus on extending the presented case study to validate the end-to-end workflow in a distributed infrastructure, where the different tasks are executed on heterogeneous systems (e.g., HPC/Cloud, CPUs/GPUs). Another path worth of investigation concerns the use of software containers for enabling fully portable workflows on different systems and the assessment of their impact on the climate simulation and processing performance.

ACKNOWLEDGMENTS

This work has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland and Norway. In Spain, it has received complementary funding from MCIN/AEI/10.13039/501100011033, Spain and the European Union NextGenerationEU/PRTR (contracts PCI2021-121957, PCI2021-121931, PCI2021-121944, and PCI2021-121927). In Italy, it has been preliminary approved for complimentary funding by Ministero dello Sviluppo Economico (MiSE) (ref. project prop. 2659). The authors also acknowledge financial support by MCIN/AEI /10.13039/501100011033, Spain through the "Severo Ochoa Programme for Centres of Excellence in R&D" under Grant CEX2021-001148-S, the Spanish Government (contract PID2019-107255 GB) and by Generalitat de Catalunya (contract 2021-SGR-00412).

REFERENCES

[1] Ivonne Anders, Karsten Peters-von Gehlen, and Hannes Thiemann. 2022. Canonical Workflows in Simulation-based Climate Sciences. *Data Intelligence*, 4, 2, (Apr. 2022), 212–225. eprint: https://direct.mit.edu/dint/article-pdf/4/2/212/2012394/dint_a_00127.pdf. DOI: 10.1162/dint_a_00127.

[2] M. Asch et al. 2018. Big data and extreme-scale computing: pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. English (US). *International Journal of High Performance Computing Applications*, 32, 4, (July 2018), 435–479. DOI: 10.1177/1094342018778123.

[3] Rosa Badia, Eduard Ayguade, and Jesus Labarta. 2017. Workflows for science: a challenge when facing the convergence of HPC and big data. *Supercomput. Front. Innov.: Int. J.*, 4, 1, (Mar. 2017), 27–47. DOI: 10.14529/jsfi170102.

[4] Rosa M Badia, Jorge Ejarque, Francesc Lordan, Daniele Lezzi, Javier Conejero, Javier Álvarez Cid-Fuentes, Yolanda Becerra, and Anna Queralt. 2019. Workflow environments for advanced cyberinfrastructure platforms. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 1720–1729. DOI: 10.1109/ICDCS.2019.00171.

[5] Rosa M. Badia, Javier Conejero, Carlos Diaz, Jorge Ejarque, Daniele Lezzi, Francesc Lordan, Cristian Ramon-Cortes, and Raul Sirvent. 2015. Comp superscalar, an interoperable programming framework. *SoftwareX*, 3-4, 32–36. DOI: 10.1016/j.softx.2015.10.004.

[6] Gordon Bell, Tony Hey, and Alex Szalay. 2009. Beyond the data deluge. *Science*, 323, 5919, 1297–1298. DOI: 10.1126/science.1170411.

[7] Jesse E. Bell et al. 2018. Changes in extreme events and the potential impacts on human health. *Journal of the Air & Waste Management Association*, 68, 4, 265–287. PMID: 29186670. DOI: 10.1080/10962247.2017.1401017.

[8] M Borodin, K De, J Garcia, Navarro, D Golubkov, A Klimentov, T Maeno, A Vaniachine, and on behalf of the ATLAS Collaboration. 2015. Scaling up atlas production system for the lhc run 2 and beyond: project prodsys2. *Journal of Physics: Conference Series*, 664, 6, (Dec. 2015), 062005. DOI: 10.1088/1742-6596/664/6/062005.

[9] Scott Callaghan et al. 2010. Scaling up workflow-based applications. *Journal of Computer and System Sciences*, 76, 6, 428–446. Special Issue: Scientific Workflow 2009. DOI: 10.1016/j.jcss.2009.11.005.

[10] Rui Chen, Weimin Zhang, and Xiang Wang. 2020. Machine learning in tropical cyclone forecast modeling: a review. *Atmosphere*, 11, 7. DOI: 10.3390/atmos11070676.

[11] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras. (2015).

[12] Matteo Coronese, Francesco Lamperti, Klaus Keller, Francesca Chiaromonte, and Andrea Roventini. 2019. Evidence for sharp increase in the economic damages of extreme natural disasters. *Proceedings of the National Academy of Sciences*, 116, 43, 21450–21455. DOI: 10.1073/pnas.1907826116.

[13] Rafael Ferreira da Silva, Ewa Deelman, Rosa Filgueira, Karan Vahi, Mats Rynge, Rajiv Mayani, and Benjamin Mayer. 2016. Automating environmental computing applications with scientific workflows. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, 400–406. DOI: 10.1109/eScience.2016.7870926.

[14] Ewa Deelman et al. 2018. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, 32, 1, 159–175. DOI: 10.1177/1094342017704893.

[15] Clara Deser et al. 2020. Insights from earth system model initial-condition large ensembles and future prospects. *Nature Climate Change*, 10, 4, 277–286. DOI: 10.1038/s41558-020-0731-2.

[16] Jorge Ejarque and Rosa M Badia. 2023. Automatizing the creation of specialized high-performance computing containers. *The International Journal of High Performance Computing Applications*, 37, 3-4, 272–287. DOI: 10.1177/10943420231165729.

[17] Jorge Ejarque, Marta Bertran, Javier Álvarez Cid-Fuentes, Javier Conejero, and Rosa M. Badia. 2020. Managing failures in task-based parallel workflows in distributed computing environments. In *Euro-Par 2020: Parallel Processing*. Maciej Malawski and Krzysztof Rzadca, (Eds.) Springer International Publishing, Cham, 411–425. ISBN: 978-3-030-57675-2. DOI: 10.1007/978-3-030-57675-2_26.

[18] Jorge Ejarque et al. 2022. Enabling dynamic and intelligent workflows for hpc, data analytics, and ai convergence. *Future generation computer systems*, 134, 414–429. DOI: 10.1016/j.future.2022.04.014.

[19] Donatello Elia, Sandro Fiore, and Giovanni Aloisio. 2021. Towards HPC and big data analytics convergence: design and experimental evaluation of a HPDA framework for eScience at scale. *IEEE Access*, 9, 73307–73326. DOI: 10.1109/ACCESS.2021.3079139.

[20] Donatello Elia, Sandro Fiore, Alessandro D'Anca, Cosimo Palazzo, Ian Foster, Dean N. Williams, and Giovanni Aloisio. 2016. An in-memory based framework for scientific data analytics. In *Proceedings of the ACM International Conference on Computing Frontiers* (CF '16). ACM, Como, Italy, 424–429. ISBN: 978-1-4503-4128-8. DOI: 10.1145/2903150.2911719.

[21] Donatello Elia, Cosimo Palazzo, Sandro Fiore, Alessandro D'Anca, Andrea Mariello, and Giovanni Aloisio. 2023. Pyophidia: a python library for high performance data analytics at scale. *SoftwareX*. [submitted].

[22] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. 2016. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9, 5, 1937–1958. DOI: 10.5194/gmd-9-1937-2016.

[23] Rafael Ferreira da Silva, Rosa Filgueira, Ilia Pietri, Ming Jiang, Rizos Sakellariou, and Ewa Deelman. 2017. A characterization of workflow management systems for extreme-scale applications. *Future Generation Computer Systems*, 75, 228–238. DOI: 10.1016/j.future.2017.02.026.

[24] S. Fiore, A. D'Anca, D. Elia, C. Palazzo, D. Williams, I. Foster, and G. Aloisio. 2014. Ophidia: a full software stack for scientific data analytics. In *2014 International Conference on High Performance Computing Simulation (HPCS)*. (July 2014), 343–350. DOI: 10.1109/HPCSim.2014.6903706.

[25] Sandro Fiore et al. 2017. Big data analytics on large-scale scientific datasets in the INDIGO-DataCloud project. In *Proceedings of the Computing Frontiers Conference* (CF'17). ACM, Siena, Italy, 343–348. ISBN: 978-1-4503-4487-6. DOI: 10.1145/3075564.3078884.

[26] Tony Hey, Stewart Tansley, and Kristin M. Tolle, (Eds.) 2009. *The impact of workflow tools on data-centric research. The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 137–145. https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/.

[27] Jeremy Goecks, Anton Nekrutenko, James Taylor, and Galaxy Team team@galaxyproject. org. 2010. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11, 1–13. DOI: 10.1186/gb-2010-11-8-r86.

[28] Jim Gray, David T. Liu, Maria Nieto-Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. 2005. Scientific data management in the coming decade. *SIGMOD Rec.*, 34, 4, (Dec. 2005), 34–41. DOI: 10.1145/1107499.1107503.

[29] R. J. Haarsma et al. 2016. High resolution model intercomparison project (highresmip v1.0) for cmip6. *Geoscientific Model Development*, 9, 11, 4185–4208. DOI: 10.5194/gmd-9-4185-2016.

[30] Tony Hey, Stewart Tansley, and Kristin Tolle. 2009. *The fourth paradigm: data-intensive scientific discovery*. Microsoft Research, (Oct. 2009). ISBN: 978-0-9825442-0-4. https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/.

[31] 2015. Indices of daily temperature and precipitation extremes. http://etccdi.pacificclimate.org/docs/ETCCDMIndicesComparison1.pdf. (2015).

[32] M. I. Jordan and T. M. Mitchell. 2015. Machine learning: trends, perspectives, and prospects. *Science*, 349, 6245, 255–260. DOI: 10.1126/science.aaa8415.

[33] Ji Liu, Esther Pacitti, Patrick Valduriez, and Marta Mattoso. 2015. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13, 457–493. DOI: 10.1007/s10723-015-9329-8.

[34] Yunjie Liu, Evan Racah, Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, and William Collins. 2016. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. (2016). arXiv: 1605.01156 [cs.CV].

[35] Domingo Manubens-Gil and Javier Vegas-Regidor. 2016. Assesment report on autosubmit, cylc and ecflow. https://earth.bsc.es/wiki/lib/exe/fetch.php?media=tools:is-enes2_d93_v1.0_mp.pdf. (2016).

[36] Martín Abadi et al. 2015. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. (2015). https://www.tensorflow.org/.

[37] Hilary J. Oliver, Matthew Shin, and Oliver Sanders. 2018. Cylc: a workflow engine for cycling systems. *Journal of Open Source Software*, 3, 27, 737. DOI: 10.21105/joss.00737.

[38] Cristian Ramon-Cortes, Albert Serven, Jorge Ejarque, Daniele Lezzi, and Rosa M Badia. 2018. Transparent orchestration of task-based parallel applications in containers platforms. *Journal of Grid Computing*, 16, 137–160. DOI: 10.1007/s10723-017-9425-z.

[39] Daniel A. Reed and Jack Dongarra. 2015. Exascale computing and big data. *Commun. ACM*, 58, 7, (June 2015), 56–68. DOI: 10.1145/2699414.

[40] S.I. Seneviratne et al. 2021. Weather and climate extreme events in a changing climate. In *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. V. Masson-Delmotte et al., (Eds.) Cambridge University Press, Cambridge, UK and New York, NY, USA. Chap. 11. DOI: 10.1017/9781009157896.013.

[41] Enric Tejedor, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M Badia, Jordi Torres, Toni Cortes, and Jesús Labarta. 2017. Pycompss: parallel computational workflows in python. *The International Journal of High Performance Computing Applications*, 31, 1, 66–82. DOI: 10.1177/1094342015594678.

[42] 2013. Topology and orchestration specification for cloud applications (tosca) primer version 1.0. https://docs.oasis-open.org/tosca/tosca-primer/v1.0/tosca-primer-v1.0.html. (2013).

[43] Pere Vergés, Francesc Lordan, Jorge Ejarque, and Rosa M. Badia. 2023. Task-level checkpointing system for task-based parallel workflows. In *Euro-Par 2022: Parallel Processing Workshops*. Jeremy Singer, Yehia Elkhatib, Dora Blanco Heras, Patrick Diehl, Nick Brown, and Aleksandar Ilic, (Eds.) Springer Nature Switzerland, Cham, 251–262. ISBN: 978-3-031-31209-0. DOI: 10.1007/978-3-031-31209-0_19.

[44] Mark D. Wilkinson et al. 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 1, 160018. DOI: 10.1038/sdata.2016.18.

[45] X. Yepes-Arbós, G. van den Oord, M. C. Acosta, and G. D. Carver. 2022. Evaluation and optimisation of the i/o scalability for the next generation of earth system models: ifs cy43r3 and xios 2.0 integration as a case study. *Geoscientific Model Development*, 15, 2, 379–394. DOI: 10.5194/gmd-15-379-2022.