Climate Data IT: Towards a National Data Hub for Climate Science

Sandro Fiore[1], Paola Nassisi[2], Ludovica Sacco[1], Gabriele Padovani[1], Fabrizio Antonio[2], Donatello Elia[2], Italo Epicoco[2,3], Nicola Giuseppe Marchioro[1], Marco Robol[1], Konstantinos Zefkilis[1], Flavio Vella[1], Carolina Sopranzetti[1], Andrea Alessandri[4], Annalisa Cherchi[4], Franco Catalano[5], Gianmaria Sannino[5], Gabriella Scipione[6], Marco Puccini[6], Erika Coppola[7], Ivan Girotto[7], Marco Reale[8], Stefano Salon[8], Antonella Galizia[9], Marcello Iotti[9], Paolo Giorgini[1], Hilary J Oliver[10], Valentine Anantharaj[11] and Silvio Gualdi[2]

[1] *University of Trento, Italy*
[2] *CMCC Foundation - Euro-Mediterranean Center on Climate Change, Italy*
[3] *Università del Salento, Lecce, Italy*
[4] *National Research Council of Italy, Institute of the Atmospheric Sciences and Climate (CNR-ISAC), Bologna, Italy*
[5] *Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile, Italy*
[6] *Cineca, Italy*
[7] *International Center for Theoretical Physics, Italy*
[8] *Istituto Nazionale di Oceanografia e di Geofisica Sperimentale - OGS, Italy*
[9] *Istituto di Matematica Applicata e Tecnologie Informatiche, Italy*
[10] *National Institute of Water and Atmospheric Research (NIWA), New Zealand*
[11] *Oak Ridge National Laboratory, Tennessee, USA*

Abstract

The increasing urgency to understand, predict, and mitigate the impact of climate change requires the integration of advanced Earth system modeling with computational and data science tools. Earth System Models (ESMs) are fundamental to this attempt, providing integrated models to simulate the interconnected behavior of atmospheric systems, oceans, hydrosphere, cryosphere, biosphere. However, the growing complexity, resolution, and computational intensity of these models require robust support from hybrid HPC systems and digital infrastructures capable of handling exascale data volumes and simulation demands. This paper highlights the work carried out within the context of the Earth & Climate spoke of the Italian National Centre on High Performance Computing, Big Data and Quantum Computing (ICSC). In particular, it describes the software infrastructure that underpins the development of Climate Data IT, which is the first attempt to create a petascale-order data platform for climate scientists. This platform serves major community experiments such as CORDEX and CMIP. According to a co-design approach, it brings together infrastructure and technology providers, as well as modeling groups, at a national level.

Keywords

Climate Data IT, Data Hub, Workflow, Provenanance

ITADATA2025: The 4th Italian Conference on Big Data and Data Science, September 9–11, 2025, Turin, Italy

sandro.fiore@unitn.it (S. Fiore)
0000-0002-8430-6087 (S. Fiore); 0000-0003-2857-4391 (P. Nassisi); 0000-0003-0235-6273 (L. Sacco); 0009-0002-3711-7643 (G. Padovani); 0000-0002-7693-0111 (F. Antonio); 0000-0002-9206-2385 (D. Elia); 0000-0002-6408-1335 (I. Epicoco); 0009-0005-1669-4451 (N. G. Marchioro); 0000-0003-4611-0371 (M. Robol); 0000-0002-5676-9228 (F. Vella); 0000-0002-2153-7961 (A. Alessandri); 0000-0002-0178-9264 (A. Cherchi); 0000-0001-9467-4687 (F. Catalano); 0000-0002-3985-9432 (G. Sannino); 0000-0001-7202-9994 (G. Scipione); 0000-0003-1769-1592 (M. Puccini); 0000-0001-6944-5815 (E. Coppola); 0000-0003-3221-5199 (I. Girotto); 0000-0001-6849-9061 (M. Reale); 0000-0003-1233-8271 (S. Salon); 0000-0003-1672-3281 (A. Galizia); 0000-0003-4627-7825 (M. Iotti); 0000-0003-4152-9683 (P. Giorgini); 0000-0002-5715-5279 (H. J. Oliver); 0000-0002-9356-1311 (V. Anantharaj); 0000-0001-7777-8935 (S. Gualdi)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-01-16

1. Introduction

Climate change and its impacts on countless sectors of society have enormously increased the demand for comprehensive scientifically robust, timely and reliable climate data for decision making to inform adaptation and mitigation policies [1]. In this context, the continued development and implementation of Earth System Models (ESMs) is key to address the complex challenges facing society in a changing climate. ESMs allow to simulate the cycling of energy, water and major geochemical elements (e.g. carbon, macronutrients) across different components of the Earth System (atmosphere, ocean, hydrosphere, cryosphere, biosphere), taking into account their major feedback mechanisms and related non-linear processes. Such approach is also useful to: i) ease the possibility to address and quantify uncertainties and errors in model parameterization that are inherent and impossible to fix in single-component approaches, and ii) improve the predictive capabilities of models, also in relationship to parameters of specific components. ESMs cover a wide range of timescales and computation requirements set by time constraints for the solution, simulation length, spatial and temporal resolution and increased complexity of the represented processes and interactions/feedbacks within the climate system. These challenges can be met only with the advanced hybrid computational systems at pre- and full exascale.

In such a scientific context the National Centre on HPC, Big Data and Quantum Computing (ICSC), provides a pivotal opportunity for the Italian scientific, industrial and economical system to address current and upcoming scientific and societal challenges, strengthening and expanding existing competences and infrastructural resources.

In particular, the *Earth & Climate* spoke (i.e., Spoke4) within ICSC aims to implement a digital infrastructure integrated in the ICSC SuperComputing facilities to: (i) assess and store robust, high–quality climatic data; (ii) facilitate the development and sharing of ESM components (e.g., models of the atmosphere, oceans, biogeochemistry, sea-ice, land-surface, vegetation, etc.); (iii) facilitate the production and management of numerical simulations; (iv) be a national asset available to the entire Italian community engaged in research, education and operational activities in the field of climate predictions and climate change, positioning our country at the forefront of climate research.

A key outcome of the project is the deployment of the Spoke4 digital infrastructure software stack on the ICSC Supercomputing facility hosted at CINECA. This activity is currently ongoing and will result in the set up of a national data hub for climate Science, known as *Climate Data IT*.

Such an effort brings together large-scale infrastructure capacity, scientific software, and big climate datasets. Most notably it provides a national hub for the entire Italian scientific community working in the climate domain.

This paper presents the main activities contributing to the implementation of the Climate Data IT national data hub. The remainder of this paper is organized as it follows: Section 2 introduces the Climate Data IT working group. Section 3 presents the general end-to-end climate scientific workflow considered in the Spoke4, whereas Section 4 deals in details with the relevant pillars of the digital infrastructure software stack, namely the *Big Data Science and learning environment* (Section 4.1) *Workflow and provenance* (Section 4.2) and *Software lifecycle management services* (Section 4.3). Section 5 draws the conclusions of this work.

2. Climate Data IT Working Group

The initial activity around the Climate Data IT project began with the formation of a working group comprising all the relevant national-level stakeholders: infrastructure providers, technology providers, and modeling groups. Throughout the project, the working group has met regularly to define requirements, discuss needs, set objectives and prioritise simulations and data delivery. A key result has been a detailed list of simulations and a related calendar delivered by the modeling groups. These cover a total of 2 Pb across 2025–2026, relating to the expected contributions for CMIP and CORDEX at the national level. It is important to note that this document has been crucial in planning many data management activities, particularly with regard to capacity planning for the storage infrastructure at CINECA.

3. End-to-End Climate Scientific Workflow

From a high-level perspective, the general climate scientific workflow that has been captured during the design stage of the Climate Data IT consists of the following three major steps:

- **Earth System Modelling**: climate models are based on well-documented physical processes to simulate the transfer of energy and materials through the climate system. Building and running a climate model is a complex process of identifying and quantifying Earth system processes, representing them with mathematical equations, setting variables to represent initial conditions and subsequent changes in climate forcing, and repeatedly solving the equations using powerful supercomputers. This step is typically related to the classical HPC community in terms of developing and running parallel ESM simulations on large-scale supercomputers. It requires a strong software stack for HPC modelling. As a result of the current extreme spatio-temporal scales, climate models simulations generate an enormous and unprecedented amount of data.
- **Big Data processing**: this step is mostly related to the processing of the big climate datasets generated during or at the end of the simulation(s) performed in the first step. It can be mostly considered as a data engineering task, where big data technologies and High Performance Data Analytics (HPDA) solutions are exploited to pre-process, transform, reduce and summarise the data. This step is at the intersection of HPC and Big data and brings advanced tools to ease the management of large datasets.
- **Data Science and learning**: this step is towards the end users, and it relates to knowledge extraction, interactive and exploratory data analysis, better understanding of the data. It leverages advanced programming tools and software ecosystems. At this stage activities are performed at the intersection of HPC, Big Data and AI.

Besides the three steps described before, it should be noted that **workflow automation** tools are key to enable seamless execution of big data pipelines, climate simulations and AI workflows; yet, a **FAIR provenance** management can track the lineage information throughout the whole end-to-end workflow, thus expanding metadata collection over the different steps and paving the way towards more advanced scenarios (i.e., computational reproducibility).

4. Main Pillars of Climate Data IT

The following subsections deal with all the relevant pillars of the Climate Data IT digital infrastructure. In particular the following topics are covered in detail:

- Big Data Science and learning environment;
- Workflow and provenance;
- Software lifecycle management services.

4.1. Big Data Science and learning environment

Over the last few years, the set of solutions for handling and processing climate data has greatly increased, providing a wider range of possibilities for supporting the study of climate change. Some solutions from the Python ecosystem targeting High Performance Data Analytics include: Xarray [2], a Python package providing support for domain specific data formats and data processing operations, Iris [3], a format-agnostic package for analysis and visualisation of Earth system data, and PyOphidia [4], the Python bindings for the Ophidia framework, targeting scalable analysis of scientific multi-dimensional data. In the meantime, Machine Learning (ML) approaches also became more common in climate applications [5]. Being able to exploit novel tools and HPC systems is key for supporting data science and scientific discoveries in climate sciences. However, the large availability of data analytics and ML solutions, together with the increasing size and complexity of HPC systems, poses also challenges in terms of efficient deployment and use of the software and computing infrastructure.

A software environment integrating HPC and data infrastructures together with cutting-edge libraries and tools can provide the means for simplifying scientists' workflow and increase their productivity for the implementation of large-scale data science and scientific artificial intelligence applications.

As an added value, a common data science and learning environment can foster a more collaborative approach and increase portability of the analysis through the use of user-friendly and self-contained documents like Jupyter Notebooks. Graphic User Interfaces (GUIs), such as the software provided by the Jupyter project, can simplify climate scientists' interactions with the computing infrastructure and represent, thus, the virtual research environment front-end for data science and learning applications in climate science.

Virtual research environments based on Jupyter project components have been explored as entry points to HPC systems, in several scientific domains, including astronomy, bioinformatics, and of course climate and Earth sciences [6, 7, 8, 9, 10, 11].

4.1.1. Architectural design

In our work, two key requirements have been considered during the design and implementation of the system:

- User-friendly environment for the development, execution and sharing of climate data science and learning applications;
- Transparent interaction with the HPC infrastructure, both in terms of authentication and resources access.

The resulting design provides a high-level overview of a climate data science and learning environment that can be customized in different ways according to the infrastructure setup and available computing capabilities. The system implementation integrates domain-specific and community-based tools and frameworks to properly address users' needs. The following diagram (Figure 1) shows the architectural design taking into account these requirements and specializing the set of technologies to those used in the climate domain.

Components from the Jupyter project and the Conda package manager can be considered capstone solutions for addressing the requirements since these allow to set up an easy-to-use software environment and simplify collaboration and sharing of the results and software configuration. The Jupyter project supports different software components for interactive analysis, data science applications development and deployment of the software on different infrastructures (both cloud and HPC). Relevant to this work is the JupyterHub, a multi-user version of the Notebook service. It represents a highly customisable component supporting different methods for user authentication and authorization (e.g., PAM, OAuth, LDAP, Kerberos, etc.) and different spawner processes for deploying the Jupyter Notebook/JupyterLab server on different infrastructures (e.g., Kubernetes, Docker, Slurm, LSF, etc.).

4.1.2. Infrastructural setup

During the project lifetime, different configurations have been tested at CMCC and UNITN with the main goal of developing the proper knowledge to easily replicate such a deployment on the target infrastructure of the ICSC Hub (i.e., CINECA).

Figure 2 shows a high-level view of the architecture diagram of the environment set up on the Juno SuperComputer at CMCC [1]. Juno has a computing power (theoretical peak performance) of about 1.1PFlops. It is a hybrid cluster based on CPUs and GPUs, composed of 170 dual-processor nodes with a total of 12,240 cores and 87 TB of main memory. The parallel file system provides around 23PB of space.

The JupyterHub instance represents the entry point to the environment and it is configured to exploit the same authentication method already used for accessing the cluster via a terminal. Jupyter servers are deployed on one of the cluster compute nodes on-demand according to user's requirements. The JupyterHub BatchSpawner, customised for the HPC environment, is used to deploy the Jupyter servers

[1] https://www.cmcc.it/what-we-do/high-performance-computing-center-hpcc

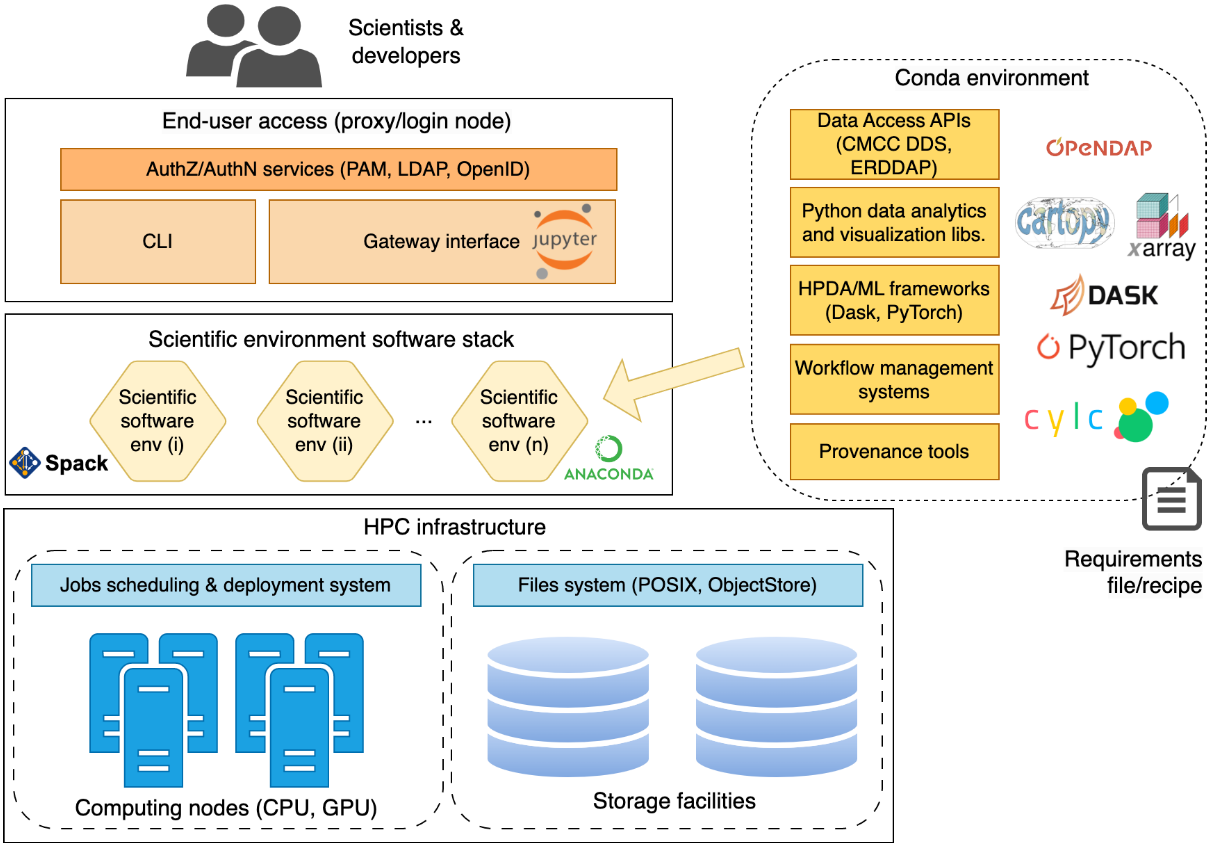

Scientists & developers
End-user access (proxy/login node)
AuthZ/AuthN services (PAM, LDAP, OpenID)
CLI
Gateway interface
jupyter
Scientific environment software stack
Scientific software env (i)
Scientific software env (ii)
...
Scientific software env (n)
Spack
ANACONDA
Conda environment
Data Access APIs (CMCC DDS, ERDDAP)
Python data analytics and visualization libs.
HPDA/ML frameworks (Dask, PyTorch)
Workflow management systems
Provenance tools
OPeNDAP
Cartopy
xarray
DASK
PyTorch
cylc
Requirements file/recipe
HPC infrastructure
Jobs scheduling & deployment system
Computing nodes (CPU, GPU)
Files system (POSIX, ObjectStore)
Storage facilities

Figure 1: General architecture design of the climate data science and learning infrastructure

on the compute nodes. In order to support various workloads and applications, different profiles are available and can be selected at the start-up of the Jupyter service. Conda environments have been pre-configured targeting some common data science and learning modules; furthermore, users can exploit one of their existing Conda envs through the ipykernel module [2].

The main advantage of such architecture is that users can run their Jupyter Notebooks directly on the HPC nodes of the cluster, exploiting the full computing and memory capacities. Users can also exploit solutions like Dask [12] for scaling processing over a large set of compute nodes, if needed. The Jupyter deployment has been operating on the Juno SuperComputer since the beginning of 2024.

In order to improve the user experience a set of extensions to the JupyterLab interface have been integrated, to support: (i) processing with a HPDA python framework, i.e. Dask, and the related extension for Dask clusters deployment and monitoring via a dashboard [3] and (ii) easy access to content (e.g., notebooks or Python code) from GitHub via another extension [4].

4.2. Workflow and provenance

Workflow and provenance play crucial roles in the experiments of climate scientists. Workflow management systems coordinate the execution of computational and data tasks on the infrastructure, while provenance involves gathering and managing the associated lineage metadata. The next subsections highlight the Climate Data IT contribution in this area, in particular, a new provenance software ecosystem developed within the project, which includes: a provenance service (Section 4.2.1), a provenance library for AI processes (Section 4.2.2), another one for provenance in workflows (Section 4.2.3) and finally a GUI to explore and navigate provenance documents (Section 4.2.4).

[2] ipykernel: https://ipython.readthedocs.io/en/5.x/install/kernel_install.html
[3] Dask JupyterLab Extension: https://github.com/dask/dask-labextension
[4] JupyterLab GitHub: https://github.com/jupyterlab/jupyterlab-github

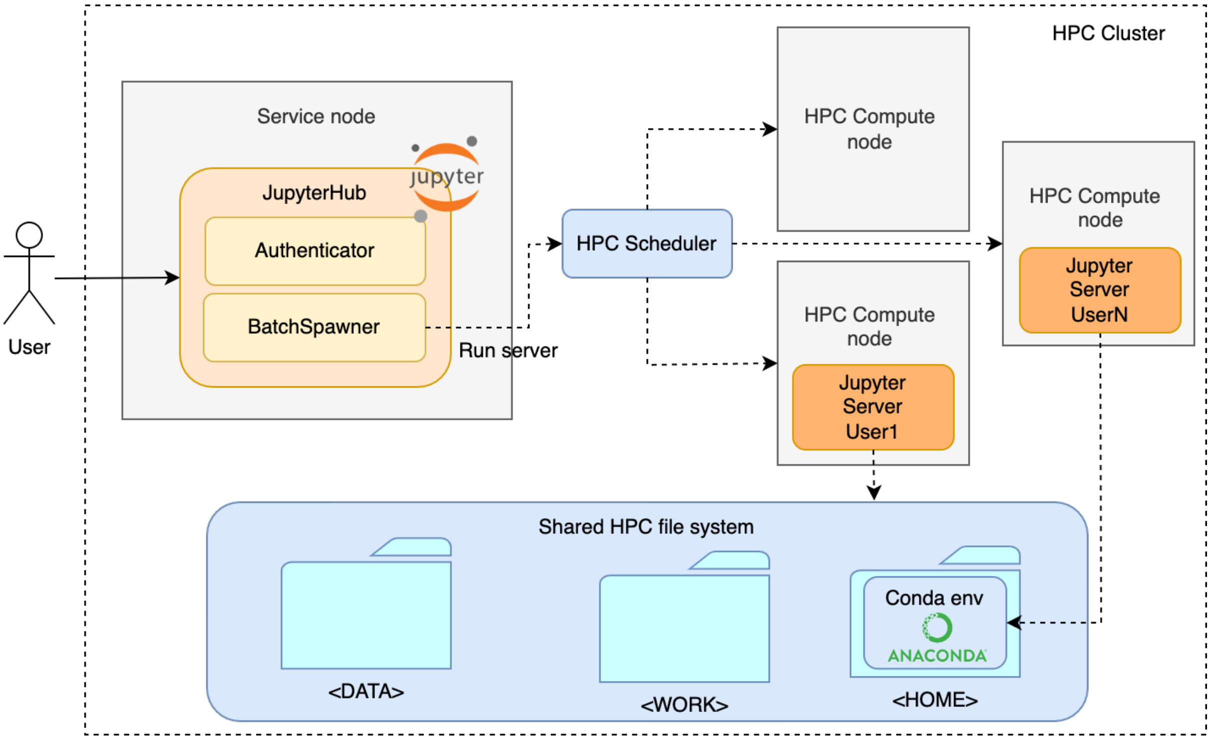


Figure 2: High-level architecture of the climate data science and learning software infrastructure deployed at CMCC

4.2.1. Provenance Service

One of the main component developed in the context of the Climate Data IT is the provenance service, which mainly provides provenance store (I/O) capabilities [13]. Its architecture adopts a straightforward web service paradigm and consists of three main components:

- a Web Service front-end, built with Python using the Flask microframework, which enables rapid development of RESTful web services and provides a robust in-memory representation of provenance documents utilizing the PROV Python library;
- the Command Line Interface (CLI), which offers a set of commands that serve as a wrapper for the RESTful API calls;
- the Graph database engine back-end, to store and query the graph elements.

The provenance service has been developed following the principles of RESTful architecture.

Five categories of resources have been identified: document, entity, activity, agent, and relation. The *document* represents the overall provenance information linked to a workflow, with each abstract document being associated one-to-one with a graph database. *Entity*, *activity*, and *agent* are sub-resources of the *document* resource, facilitating CRUD operations on the three types of elements stored in a provenance graph based on the W3C PROV data model. This ensures precise control over each node stored in the graph database. Similarly, the *relation* resource enables actions on each individual relationship within the graph.

4.2.2. Provenance Data Collection for AI Tasks

It is not a novel concept anymore that the field of machine learning has experienced an exponential speedup in both the development and publishing of new works. While this rapid pace of research has undoubtedly brought many benefits, it has also led to an increasing amount of work that is conducted with little rigor and in a superficial manner. Undocumented code and non-reproducible results inevitably lead to confusion among researchers and an environment where trust is not at the core of the work being proposed. Keeping a detailed record of the entire design process is essential for fully reproducing experiments and avoiding unproven results. In addition to these difficulties, there is no straightforward method for determining the value of the many hyperparameters used in the training of machine learning models. By collecting multiple sets of experiments, users could look at similar goals and identify hyperparameter values that might be ideal for their application. Another problem with hyperparameter tuning is the repeated attempts to train the optimal model. When the same process is repeated several times, a significant amount of computational resources are wasted. Given the large size of many machine learning models, this paradigm quickly becomes unsustainable, especially when dealing with architectures consisting of billions of parameters. To this end, the yProv4ML[5] library [14] exposes logging utilities providing a recognisable interface for storing provenance data. It collects three main categories of information: artifacts, parameters and metrics, of which, the latter identifies any file or output that may be used later in the next phases of the workflow. For machine learning processes, these typically include model versions, checkpoints and source code material. Parameters, on the other hand, are one-time recorded values used during training. Some examples are the learning rate, model size or width, and the optimizer used. The last category contains information that is updated during the training process. These metrics include losses and program execution statistics such as energy efficiency, power consumption and GPU usage. Once information about a single run is stored, it is also possible to compare the results of successive, related executions. This allows a better understanding of the impact of hyperparameters and model configurations, while keeping track of any changes to the overall script. yProv4ML implements an ad hoc data model to store all the information collected during program execution in a memory-efficient manner. In particular a caching system automatically decides when it is more opportune to empty the values stored in RAM memory, avoiding slowdowns of the training process. The library is also capable of constructing a provenance graph, which contains all the data that has been stored during the execution of the program, both in dot and in svg format. The usefulness of yProv4ML can be apparent when monitoring processes with high energy consumption, where a preliminary testing phase could be used to understand which part of the program causes the most impact, and consequently reduce the overall consumption when the real training phase is run. The ability to log multiple versions of the same experiment is also critical to understanding which hyperparameters work better with the current execution, and to avoid repeating the same errors across multiple runs. The importance of applying provenance tracking is demonstrated by the broad number of tools developed by the community in recent years, tools whose purpose is to collect information during the execution of a training workflow and then present it to the user, allowing analysis and considerations on the results and the process that led to their generation. The W3C has developed a standard, called PROV, for collecting provenance information of generic systems without referring to specific cases of Machine Learning; precisely because of this generality, specific language expansions for ML have been proposed [15]. However, the tools that produce a PROV representation manage the entire learning process, making it necessary to revolutionize the entire workflow if one decides to adopt them, thereby discouraging the adoption itself and, in general, the change of tools. This situation opens up space for new tools capable of converting a native representation into a PROV representation in a transparent manner for the user. The JSON format is a widely used one for representing data, and being a text format, makes writing, modifying and reading easier. For this reason, the yProv4ML library was designed to save information in JSON format [16], and allows for the creation of a provenance graph and its visual representation.

To validate the functionalities of yProv4ML, we set to benchmark Prithvi-EO [17], a vision transformer-based foundation model, for image reconstruction. Accurate reconstruction of finer scale earth surface features from coarse-resolution data is a critical challenge in geoscientific modeling, as well as a benchmark for understanding the performance of environmental models. We implemented the Prithvi-EO 300M parameter architecture, trained on 4.2M time series samples from Sentinel 2 [18], and run fine-tuning on the Moderate Resolution Imaging Spectroradiometer (MODIS) dataset [19], for masked image reconstruction. We fine-tune the model on a subset of 25.000 128x128 patches and validate its

[5] https://github.com/HPCI-Lab/yProvML

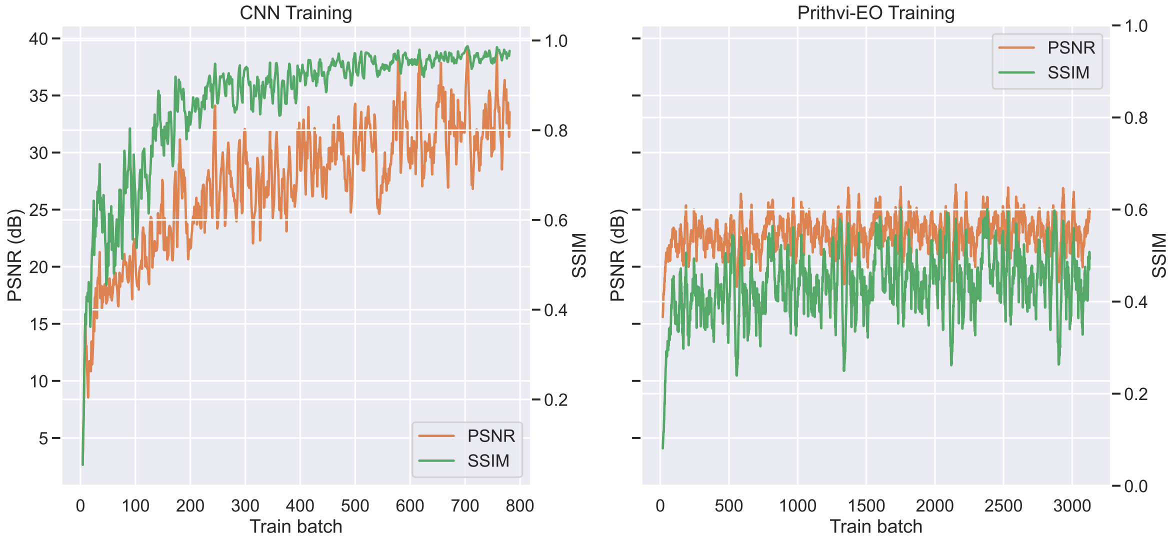


Figure 3: Quantitative metrics: Peak Noise to Signal Ratio and Structural Similarity Index Measure comparison for SR-CNN (left) and Prithvi-EO (right).

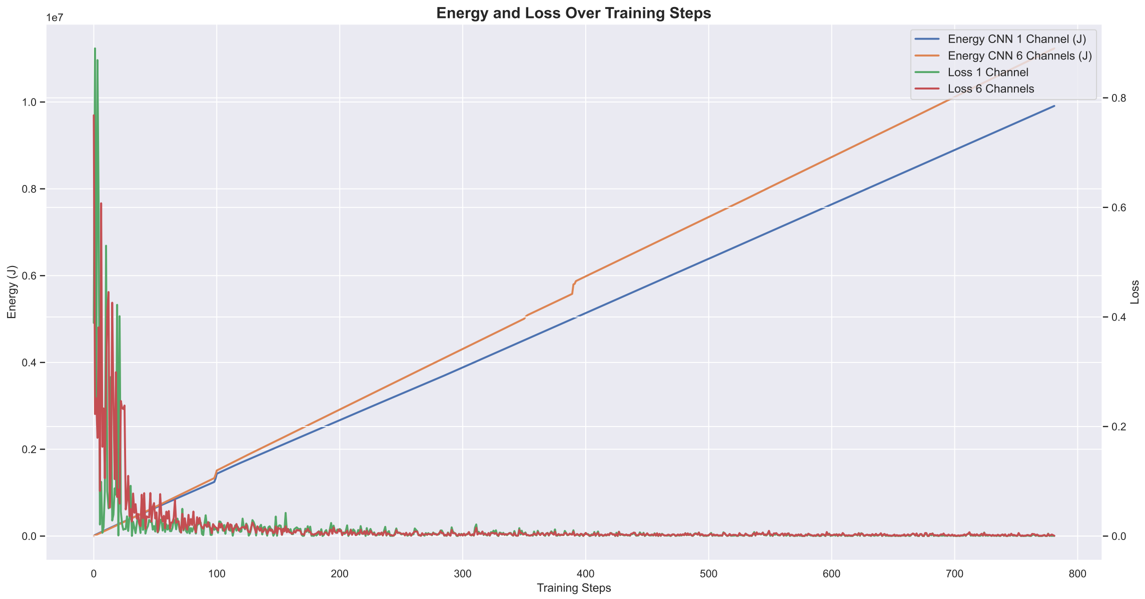


Figure 4: Loss relation to energy consumption for two configurations of SR-CNN, one trained on, respectively, the first one the first six bands of the MODIS dataset.

output on MODIS tiles. Subsequently we fully train a set of super-resolution CNN based ML models [20], whose metrics are shown in Figure 4, on the same input and output domain, to mirror the approach on the ViT architecture, and the results both from a performance and energy perspective are compared. Both the models are trained with DDP strategy [21] on 4 nodes of the Frontier supercomputer, each equpped with 8 GPUs. We focused on DDP since both models fit entirely on one node, and as it is easier to have them converge on limited data.

Our aim is to highlight the discrepancy in performance between the ad-hoc solution and the general purpose foundation model. The energy requirements of foundation models, shown in Figure 3, specifically in the pre-training phase, is compared with simpler architectures, which may offer similar performance but at a fraction of the cost. Even when passing more data to the Prithvi-EO model, up to

80000 samples for 5 epochs, the resulting PSNR raises only to ∼ 0.7 and SSIM to ∼ 0.65, indicating a lack of compute time and data. It remains clear, from the results shown, that foundation models guarantee better generalization and adaptability to data domains beyond the pretraining distribution, and that they tend to be much more energy efficient when having to scale to large use cases. It is however also apparent, that a smaller, ad-hoc solution is still to be preferred in situations where the use cases is not excessively complex.

4.2.3. Provenance service for workflow management systems

End-to-end ESMs workflows often consist of numerous tasks executed across heterogeneous computing environments, such as running ESM models on high-performance computing (HPC) clusters, artificial intelligence (AI) tasks on GPU-based systems, and data analytics on cloud infrastructures. In such contexts, tracking provenance in large-scale computational workflows is essential to ensure the reproducibility and reliability of research. Scientific experiments, especially those involving complex simulations and large datasets, require thorough documentation of every phase, from data acquisition to simulation execution and result analysis. Following an initial phase of technology scouting and preliminary analysis of integrated workflow-provenance scenarios, development efforts have been directed toward implementing a specific library, called yProv4WFs and designed to enable provenance tracking within Workflow Management Systems (WfMSs) [22] [23]. According to the outcomes of an initial end users consultation, Streamflow [24] and Cylc [25] have been elected as the first two candidates to be studied and integrated in yProv4WFs. Cylc is a general purpose workflow engine optimized for orchestrating cycling systems. Widely used in operational weather, climate and environmental forecasting on HPC platforms, it is domain-agnostic and capable of automatically executing tasks based on defined schedules and dependencies. In Cylc workflows, each step typically corresponds to a script or application that performs a discrete computational task. StreamFlow is an innovative approach to workflow execution that combines workflow graphs with the description of potentially complex execution environments that do not necessarily share a common data space. Based on the Common Workflow Language (CWL) standard, it supports concurrent task execution in multi-agent, multi-container ecosystems. The yProv4WFs library is a key part of the yProv software ecosystem [26], designed to address the challenges of managing multi-level provenance and ensuring reproducibility in end-to-end scientific workflows. It serves as a third-party tool for tracking data provenance across various WfMSs. By providing a standardised and shareable provenance framework, yProv4WFs enables users from different systems, working on the same project, to understand and utilise provenance data without requiring deep knowledge of each other's workflows. This facilitates better collaboration, transparency, and reproducibility across multiple platforms. yProv4WFs adheres to the W3C PROV standards, ensuring that the provenance data it generates is consistent with globally recognized norms. Compliance with W3C PROV not only enhances the utility and flexibility of provenance data, but also ensures compatibility with other tools and systems supporting the same standards, thereby extending its relevance, FAIRness and usefulness over time. Table 1 highlights the connection between the W3C PROV standard and the yProv4WFs library. Each row of the table corresponds to a term from the PROV-O ontology and identifies its counterpart or equivalent in the yProv4WFs model. This alignment demonstrates both conformity to established standards and the capacity of yProv4WFs to represent the complexity of the provenance of scientific workflows with greater precision. The library is capable of tracking both simple (pipeline) and complex constructs (cycles) that adapt to the needs of sophisticated, large-scale workflows.

4.2.4. Provenance Explorer

Our contribution fills the gap between existing libraries and services and an application designed to increase the final user experience. To do this, it has been implemented the yProvExplorer, a Scientific Gateway that addresses the user's need to navigate complex provenance documents, providing a nicely organized representation of all the interconnected provenance elements. All such elements are arranged

Table 1
How yProv4WFs components map to and integrate W3C PROV-O standards

W3C PROV-O	yProv4WFs

prov:Activity	`Node`
	`Workflow`
	`Task`

prov:Entity	`Data`
prov:Agent	`Agent`
prov:startedAtTime	_start_time → attribute class `Node`
prov:endedAtTime	_end_time → attribute class `Node`
prov:used	add_input() → method class `Workflow, Task`
prov:wasGeneratedBy	add_input() → method class `Workflow, Task` set_consumer() → method class `Data`
prov:wasDerivedFrom (represents the dependencies output → input)	set_consumer() → method class `Data` set_producer() → method class `Data`
prov:wasInformedBy (dependencies next_task → wasInformedBy → task)	set_prev() → method class `Node` set_next() → method class `Node`
prov:actedOnBehalfOf	set_acted_for() → method class `Agent`
prov:wasAssociatedWith (from prov:Activity to prov:Agent)	set_agent() → method class `Node` associated_with → list class `Agent`
prov:wasAttributedTo (from prov:Entity to prov:Agent)	set_agent() → method class `Data` attributed_to → list class `Agent`

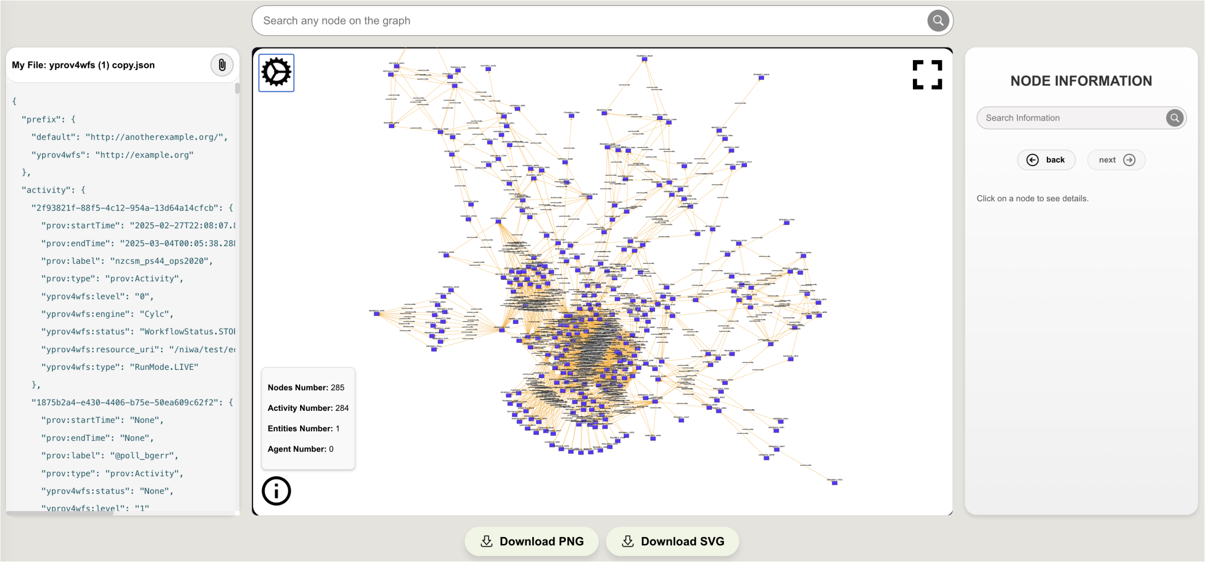
in a 2-dimensional space using a force-directed graph drawing algorithm[6], provided by the library D3js[7]. The yProvExplorer consists of a self-contained web application implemented in React[8] that can be easily deployed through Docker[9]. It nicely integrates with the yProv ecosystem, therefore, allowing to load the documents directly from yProv storage service, in addition to open document directly from the local device. The tool guides the user in navigating the graph through links to the parent and child nodes, allowing to easily move toward the source data or toward the final data. Figure 5 illustrates the yProvExplorer, which visualizes the provenance output generated during the execution of a complex workflow in Cylc. Through the yProvExplorer interface, a JSON file can be uploaded to display the corresponding provenance graph.

4.3. Software lifecycle management services

The software lifecycle provides a structured approach to improving the development process of software, ensuring the implementation, maintenance, replacement, and enhancement of specific components. Version control is a core best practice in modern software development. While it has not yet achieved universal adoption in scientific coding, its usage has grown substantially in recent years.

In the context of Spoke4, following an initial exploratory phase, the decision was made to set up a managed service leveraging GitHub's functionalities and the concept of GitHub Organisations. This workspace has been instrumental in delivering the software counterpart, the *Climate Software Hub*[10], of the Climate Data IT. This activity has fostered the use of these collaborative tools and best practices in the community for software development; it has strongly contributed to a better sharing of software and software development process; it has created a central repository at national level as well as a reference

[6] https://en.wikipedia.org/wiki/Force-directed graph drawing
[7] D3 https://d3js.org/
[8] https://react.dev/
[9] https://www.docker.com/
[10] https://github.com/ICSC-CN-HPC-Spoke-4-Earth-Climate.



My File: yprov4wfs (1) copy.json

```
{
  "prefix": {
    "default": "http://anotherexample.org/",
    "yprov4wfs": "http://example.org"
  },
  "activity": {
    "2f93821f-88f5-4c12-954a-13d64a14cfcb": {
      "prov:startTime": "2025-02-27T22:08:07.
      "prov:endTime": "2025-03-04T00:05:38.28
      "prov:label": "nzcsm_ps44_ops2020",
      "prov:type": "prov:Activity",
      "yprov4wfs:level": "0",
      "yprov4wfs:engine": "Cylc",
      "yprov4wfs:status": "WorkflowStatus.STO
      "yprov4wfs:resource_uri": "/niwa/test/e
      "yprov4wfs:type": "RunMode.LIVE"
    },
    "1875b2a4-e430-4406-b75e-50ea609c62f2": {
      "prov:startTime": "None",
      "prov:endTime": "None",
      "prov:label": "@poll_bgerr",
      "prov:type": "prov:Activity",
      "yprov4wfs:status": "None",
      "yprov4wfs:level": "1"
```

Nodes Number: 285
Activity Number: 284
Entities Number: 1
Agent Number: 0

Figure 5: Visualization of a provenance graph generated during the execution of a complex workflow in Cylc

software hub for climate scientists.

5. Conclusions

The National Centre on HPC, Big Data and Quantum Computing (ICSC), through its Earth & Climate spoke, offers a digital infrastructure to support large-scale simulations and to manage the rapidly growing volumes of climate data. The ICSC also supports the climate research community by providing a flexible, modular, and interoperable software ecosystem for simulation and data analysis. A digital infrastructure for climate Science integrates HPC systems, data infrastructures, and data science libraries, supporting reproducibility, enhancing portability, and fostering collaborative workspaces has been set up. Provenance tracking systems have been proposed, ensuring traceability, reproducibility, and transparency of scientific experiments by managing computational tasks and preserving detailed metadata on their execution. The development of such a comprehensive provenance ecosystem (a dedicated provenance service, libraries for AI processes and workflows, and a GUI-based exploration tool) represents a significant step forward in the management of scientific workflows. Furthermore, modern software development practices, such as the use of structured software lifecycles and version control systems like GitHub, have been adopted to improve code quality, facilitate collaboration, and ensure long-term maintainability, while also enabling coordinated contributions across distributed teams and projects. Taken all these efforts together, Climate Data IT represents a significant advancement in the scientific, technological, and collaborative capacity of Italy's climate research ecosystem. The integrated approach presented in the paper improves the capacity to deliver actionable climate intelligence for societal adaptation and resilience in the face of ongoing and future climate change challenges.

Acknowledgments

This work was funded under the NRRP, Mission 4 Component 2 Investment 1.4, by the European Union – NextGenerationEU (proj. nr. CN_00000013).

Valentine Anantharaj is supported by the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] C. P. Weaver, R. J. Lempert, C. Brown, J. A. Hall, D. Revell, D. Sarewitz, Improving the contribution of climate model information to decision making: the value and demands of robust decision frameworks, WIREs Climate Change 4 (2013) 39–60. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcc.202. doi:https://doi.org/10.1002/wcc.202. arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcc.202.

[2] S. Hoyer, J. Hamman, xarray: N-D labeled arrays and datasets in python, Journal of Open Research Software 5 (2017). URL: http://doi.org/10.5334/jors.148. doi:10.5334/jors.148.

[3] Iris contributors, Iris, ???? URL: https://github.com/SciTools/iris. doi:10.5281/zenodo.595182.

[4] D. Elia, C. Palazzo, S. Fiore, A. D'Anca, A. Mariello, G. Aloisio, Pyophidia: A python library for high performance data analytics at scale, SoftwareX 24 (2023) 101538. URL: https://www.sciencedirect.com/science/article/pii/S2352711023002340. doi:https://doi.org/10.1016/j.softx.2023.101538.

[5] C. Irrgang, N. Boers, M. Sonnewald, E. A. Barnes, C. Kadow, J. Staneva, J. Saynisch-Wagner, Towards neural earth system modelling by integrating artificial intelligence in earth system science, Nature Machine Intelligence 3 (2021) 667–674.

[6] A. Kanterakis, N. Karacapilidis, L. Koumakis, G. Potamias, On the development of an open and collaborative bioinformatics research environment, Procedia Computer Science 126 (2018) 1062 – 1071. URL: https://doi.org/10.1016/j.procs.2018.08.043. doi:10.1016/j.procs.2018.08.043.

[7] T. P. Maxwell, D. Duffy, L. Carriere, G. L. Potter, The Earth Data Analytic Services (EDAS) Framework, in: AGU Fall Meeting Abstracts, volume 2018, 2018, pp. IN53D–0649.

[8] M. B. Milligan, Jupyter as common technology platform for interactive HPC services, in: Proceedings of the Practice and Experience on Advanced Research Computing, PEARC '18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3219104.3219162. doi:10.1145/3219104.3219162.

[9] D. Yin, Y. Liu, H. Hu, J. Terstriep, X. Hong, A. Padmanabhan, S. Wang, CyberGIS-Jupyter for reproducible and scalable geospatial analytics, Concurrency and Computation: Practice and Experience 31 (2019) e5040. URL: https://doi.org/10.1002/cpe.5040. doi:10.1002/cpe.5040, e5040 cpe.5040.

[10] S. Cholia, L. Heagy, M. Henderson, D. Paine, J. Hays, L. Bianchi, D. Ghoshal, F. Pérez, L. Ramakrishnan, Towards interactive, reproducible analytics at scale on HPC systems, in: 2020 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC), 2020, pp. 47–54. URL: https://doi.org/10.1109/UrgentHPC51945.2020.00011. doi:10.1109/UrgentHPC51945.2020.00011.

[11] S. Juneau, K. Olsen, R. Nikutta, A. Jacques, S. Bailey, Jupyter-enabled astrophysical analysis using data-proximate computing platforms, Computing in Science Engineering 23 (2021) 15–25. doi:10.1109/MCSE.2021.3057097.

[12] M. Rocklin, Dask: Parallel computation with blocked algorithms and task scheduling, in: K. Huff, J. Bergstra (Eds.), Proceedings of the 14th Python in Science Conference, SciPy, Austin, TX, USA, 2015, pp. 130 – 136. URL: https://doi.org/10.25080/Majora-7b98e3ed-013. doi:10.25080/Majora-7b98e3ed-013.

[13] S. Fiore, M. Rampazzo, D. Elia, L. Sacco, F. Antonio, P. Nassisi, A graph data model-based micro-provenance approach for multi-level provenance exploration in end-to-end climate workflows, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 3332–3339. doi:10.1109/BigData59044.2023.10386983.

[14] G. Padovani, V. Anantharaj, S. Fiore, yprov4ml: Effortless provenance tracking for machine

learning systems, SoftwareX 31 (2025) 102298. URL: https://www.sciencedirect.com/science/article/pii/S235271102500264X. doi:https://doi.org/10.1016/j.softx.2025.102298.

[15] R. Souza, L. Azevedo, V. Lourenço, E. Soares, R. Thiago, R. Brandão, D. Civitarese, E. Brazil, M. Moreno, P. Valduriez, et al., Provenance data in the machine learning lifecycle in computational science and engineering, in: 2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), IEEE, 2019, pp. 1–10.

[16] T. D. Huynh, M. O. Jewell, A. Sezavar Keshavarz, D. T. Michaelides, H. Yang, L. Moreau, The prov-json serialization (2013).

[17] D. Szwarcman, S. Roy, P. Fraccaro, Þ. E. Gíslason, B. Blumenstiel, R. Ghosal, P. H. de Oliveira, J. L. d. S. Almeida, R. Sedona, Y. Kang, et al., Prithvi-eo-2.0: A versatile multi-temporal foundation model for earth observation applications, arXiv preprint arXiv:2412.02732 (2024).

[18] D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, M. Ranagalage, Sentinel-2 data for land cover/use mapping: A review, Remote sensing 12 (2020) 2291.

[19] C. O. Justice, E. Vermote, J. R. Townshend, R. Defries, D. P. Roy, D. K. Hall, V. V. Salomonson, J. L. Privette, G. Riggs, A. Strahler, et al., The moderate resolution imaging spectroradiometer (modis): Land remote sensing for global change research, IEEE transactions on geoscience and remote sensing 36 (1998) 1228–1249.

[20] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, A. R. Ganguly, Deepsd: Generating high resolution climate change projections through single image super-resolution, in: Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining, 2017, pp. 1663–1672.

[21] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, et al., Pytorch distributed: Experiences on accelerating data parallel training, arXiv preprint arXiv:2006.15704 (2020).

[22] L. Sacco, C. Sopranzetti, S. Fiore, Enabling provenance tracking in workflow management systems, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 4402–4409. doi:10.1109/BigData62323.2024.10825405.

[23] H. Omidi, L. Sacco, V. Hutter, G. Irsiegler, M. Claus, M. Schobben, A. Jacob, M. Schramm, S. Fiore, Towards provenance-aware earth observation workflows: the openeo case study, in: 21st IEEE International eScience Conference, 2025, to appear.

[24] I. Colonnelli, B. Cantalupo, I. Merelli, M. Aldinucci, StreamFlow: cross-breeding cloud with HPC, IEEE Transactions on Emerging Topics in Computing 9 (2021) 1723–1737. URL: https://doi.org/10.1109/TETC.2020.3019202. doi:10.1109/TETC.2020.3019202.

[25] H. J. Oliver, M. Shin, O. Sanders, Cylc: A workflow engine for cycling systems, Journal of Open Source Software 3 (2018) 737. URL: https://doi.org/10.21105/joss.00737. doi:10.21105/joss.00737.

[26] G. Padovani, V. Anantharaj, L. Sacco, T. Kurihana, M. Bunino, K. Tsolaki, M. Girone, F. Antonio, C. Sopranzetti, M. Fronza, S. Fiore, A software ecosystem for multi-level provenance management in large-scale scientific workflows for ai applications, in: SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2024, pp. 2024–2031. doi:10.1109/SCW63240.2024.00253.