# Bridging Cloud and HPC for Operational Ocean Forecasting

**Jacopo Nespolo, PhD**

Funded by the European Union
NextGenerationEU

HPC-AI Swiss Conference
Locarno 23 April 2026

exact lab
Solutions for the digital revolution

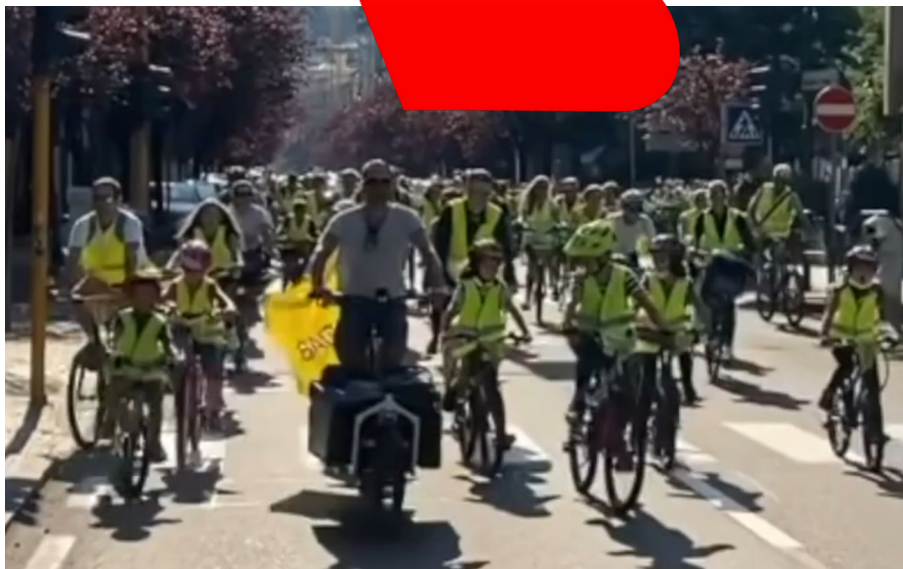
# Cloud / HPC

Two worlds that weren't designed to meet...

...until operational deadlines forced the issue.

# Hi!

- Computational / statistical physicist by education
- 4.5y postdoctoral research in quantum phase transitions and quantum optics in solid state
- Scientific software architect and developer by trade
- Bike mechanic by passion



**In eXact lab**
Since 2021 as sci-dev
Head of innovation since 2024

**In the MER project**
Technical lead / architect

exact lab

# Today's menu

1. Operational oceanography and the MER project
2. IT Challenges of operational oceanography
3. Case study: **hybrid cloud-HPC architecture**

I grew up here :)

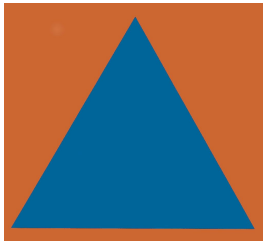
# Operational oceanography and the MER project

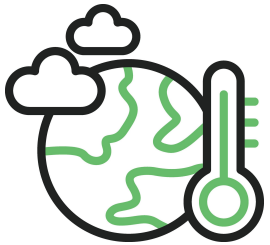
# Operational oceanography (OO)

The systematic collection, analysis, and forecasting of **ocean data**.

Supports practical applications like navigation, climate monitoring, fisheries, and public safety.

Think **weather forecasting** but for the **sea**





# OO and the MER project

MARINE ECOSYSTEM RESTORATION

## Marine Ecosystem Restoration

- 400M€ NextGenEU project led by **ISPRA** (Italian environmental protection agency)
- restore and protect marine and coastal ecosystems
- assess environmental status of Italian seas
- strengthens **operational oceanography** to improve forecasting, coastal risk management, and sustainable marine use

ISPRA
Istituto Superiore per la Protezione e la Ricerca Ambientale

Sistema Nazionale per la Protezione dell'Ambiente

**Funded by the European Union**
NextGenerationEU

**In collaboration with:**

OGS
Istituto Nazionale di Oceanografia e di Geofisica Sperimentale

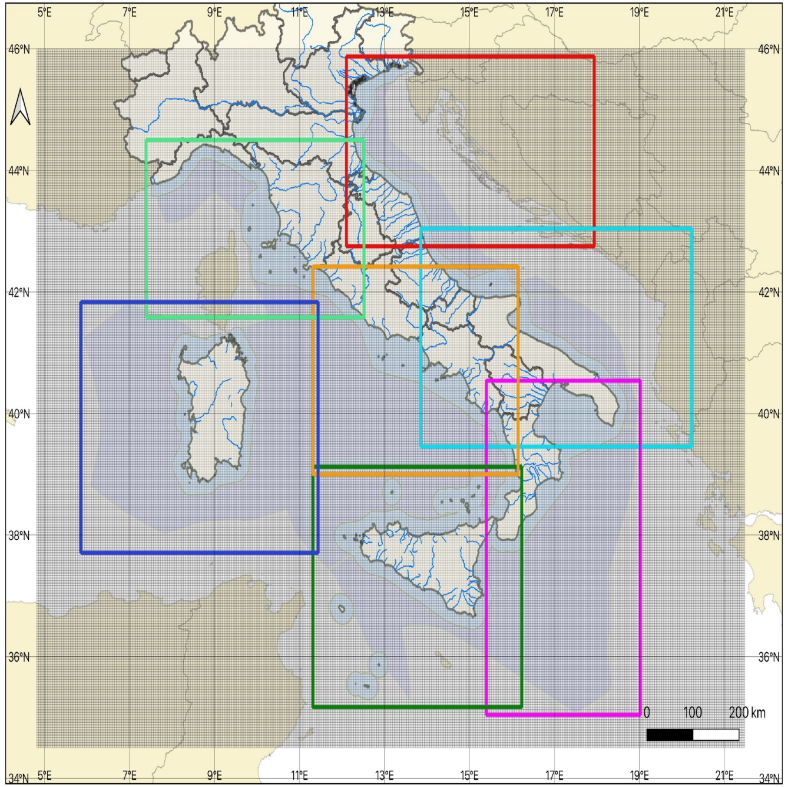
CINECA

# OO and the MER project

## Our work

Devise a system to:

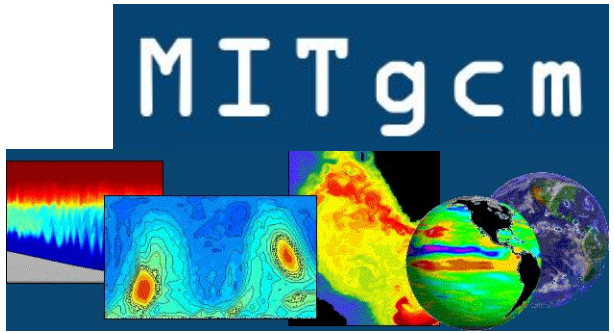
- Run marine forecasts
  - Including biogeochemistry
- Across **7 high resolution sites** covering the Italian seas (~500m)
- Across **10 very high resolution sites** of interest (~100m)
- Producing **~10TB of data** per day
- Consuming **~20k core-h** per day



# OO and the MER project

## NOT our work

The actual modelling.
Many thanks to our colleagues at




OGS
Istituto Nazionale
di Oceanografia
e di Geofisica
Sperimentale

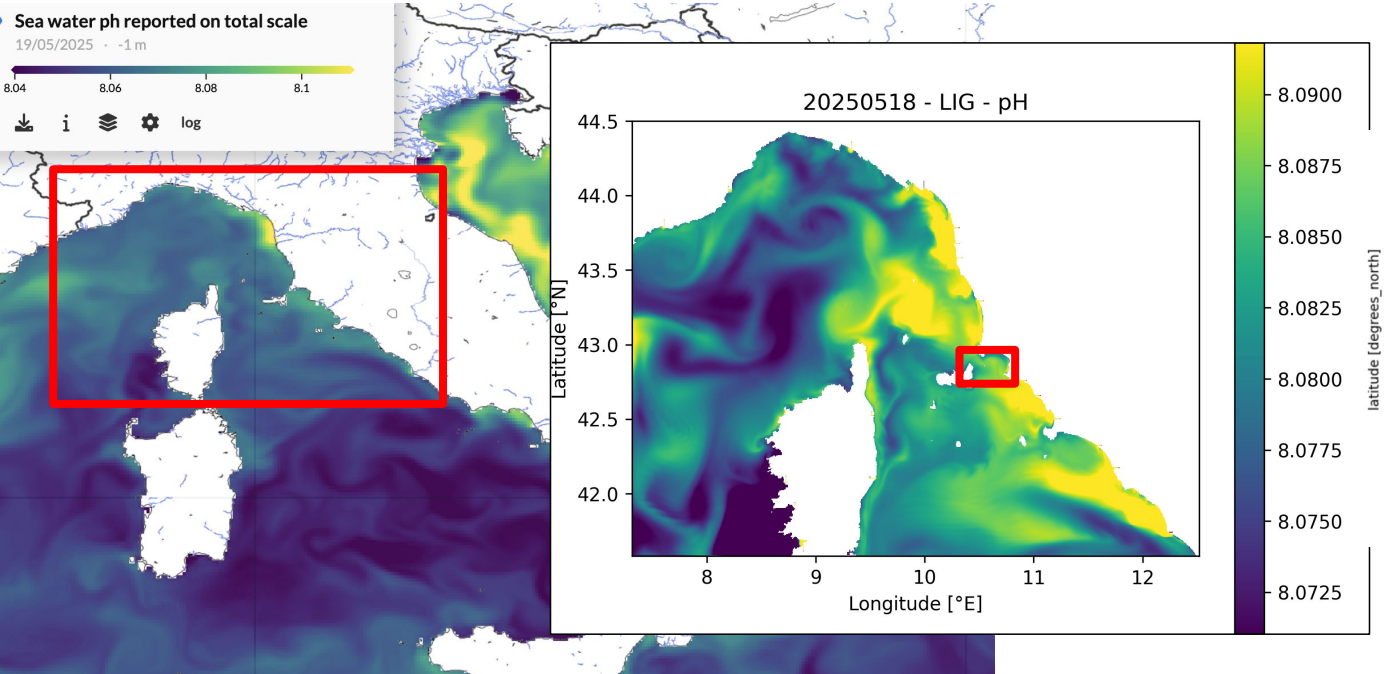
BFM
BIOGEOCHEMICAL FLUX MODEL

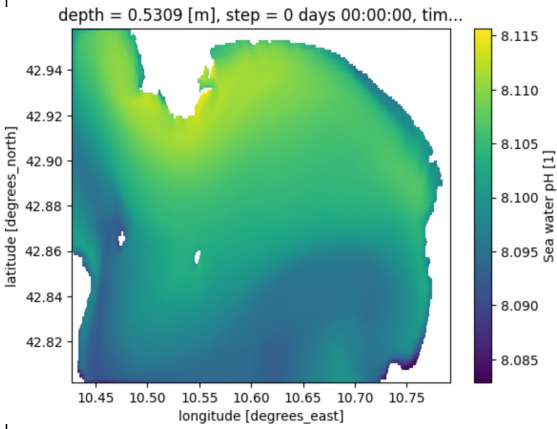
# Results

Copernicus Marine Service ~4km (Med-wide)

HR ~500m Ligurian sea

VHR ~100m Gulf of Follonica



# IT challenges

# Theory and practice

## In theory

Textbook ETL pattern

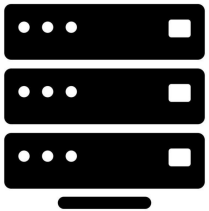
1. Fetch inputs
2. Compute forecast
3. Serve outputs

## In practice

1. Fetch
   - Some nodes are isolated from the internet
   - Each data source uses its own format + protocol
2. Compute
   - Secure access to compute resources
   - Make compiling the model reproducible (as possible)
   - Handle failures
3. Serve outputs
   - Try to move as little data as possible

\+ **Lots** of legacy code, notifications, fallbacks, monitoring, etc.

# Additional requirements



- High computing performance
- Parallel block filesystems (favours large files)
- Strict access controls
- Workload management

Ingest large input data

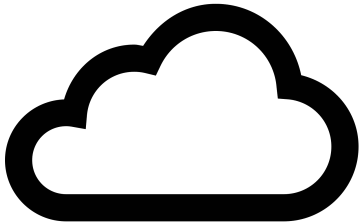
Data ready by 12:00 UTC

Monitoring

Observability

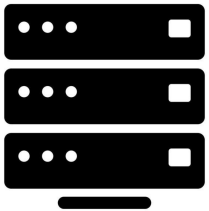
Notifications / fallbacks

Serving output data



- Low computing performance
- Ideally object storage (favours small files)
- Allows serving data
- Workflow management

# Additional requirements



Ingest large input data

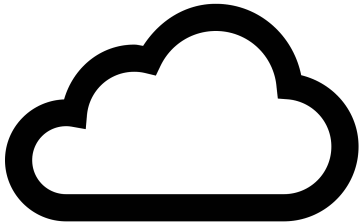
Data ready by 12:00 UTC

Monitoring

Observability

Notifications / fallbacks

Serving output data



- High computing performance
- Parallel filesystems (favours large files)
- Strict access controls
- Workload management

- Low computing performance
- Ideally object storage (favours small files)
- Allows serving data
- Workflow management

# Networking (and lack thereof)

## Compute nodes are isolated from the Internet

That’s a serious problem when you need to download several TB of input data each morning.

Downloads must take place either on the login nodes or in the cloud.

# Data sources

## No shared standard

Each data source requires ad-hoc pipelines.










# Security

## Access to Cineca Leonardo

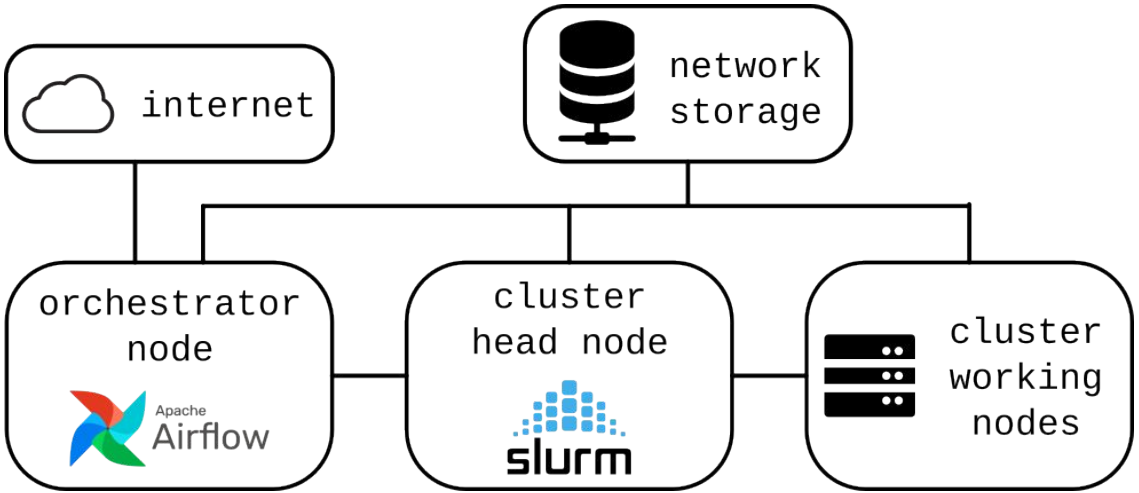
The Leonardo supercomputer is considered a critical infrastructure, mandating increased cybersecurity standards.

We had to **VAPT-certify**[1] the cloud segment of our infrastructure to obtain automated access lasting >48h.

[1] Vulnerability Assessment and Penetration Testing

# Case study:
# hybrid cloud-HPC architecture

# Our solution



# A complete re-engineering

**Declarative**
Describe simulation domains in plain files
**Modular**
Use as library or orchestrated
**Automated**
Deploying should not require HPC skills
**Modern**
Renew the tech stack, embrace standards

**60+ workflow tasks**

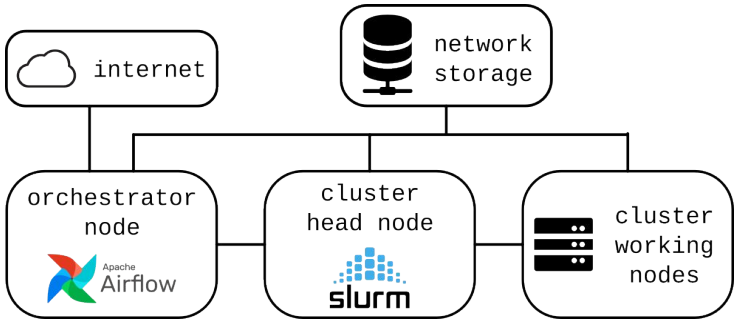
**25+ kLoC Python**

**~300 unit and integration tests**

# Cloud segment - VPS

## Apache Airflow + PostgreSQL

Deployed in containers on two separate VPS hosted on Cineca's ADACloud Openstack installation.




## Cloud storage is separate from HPC!

We had to resort to `sshfs` to mount the HPC working directories.

openstack

OPEN CONTAINER INITIATIVE

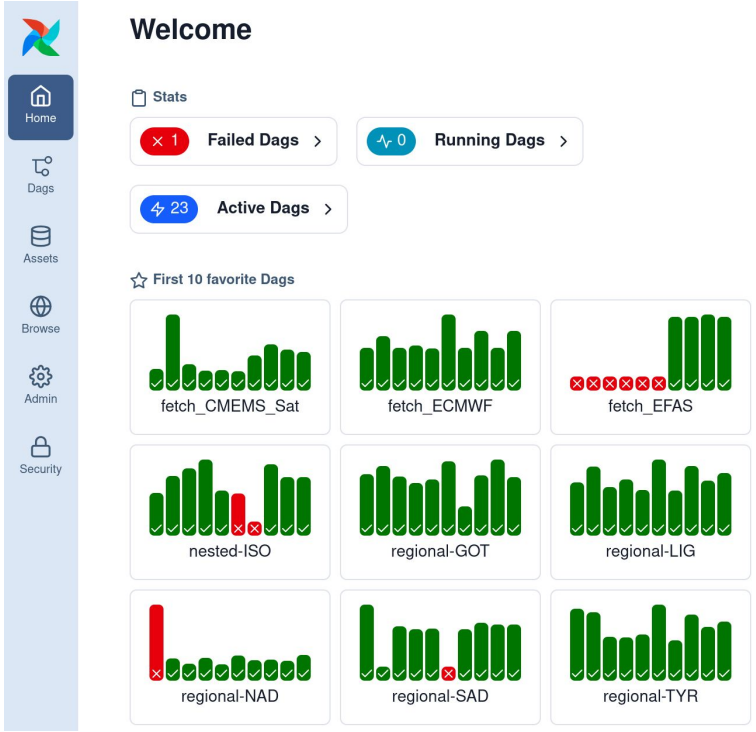
# Cloud segment - Apache Airflow

## Workflow manager

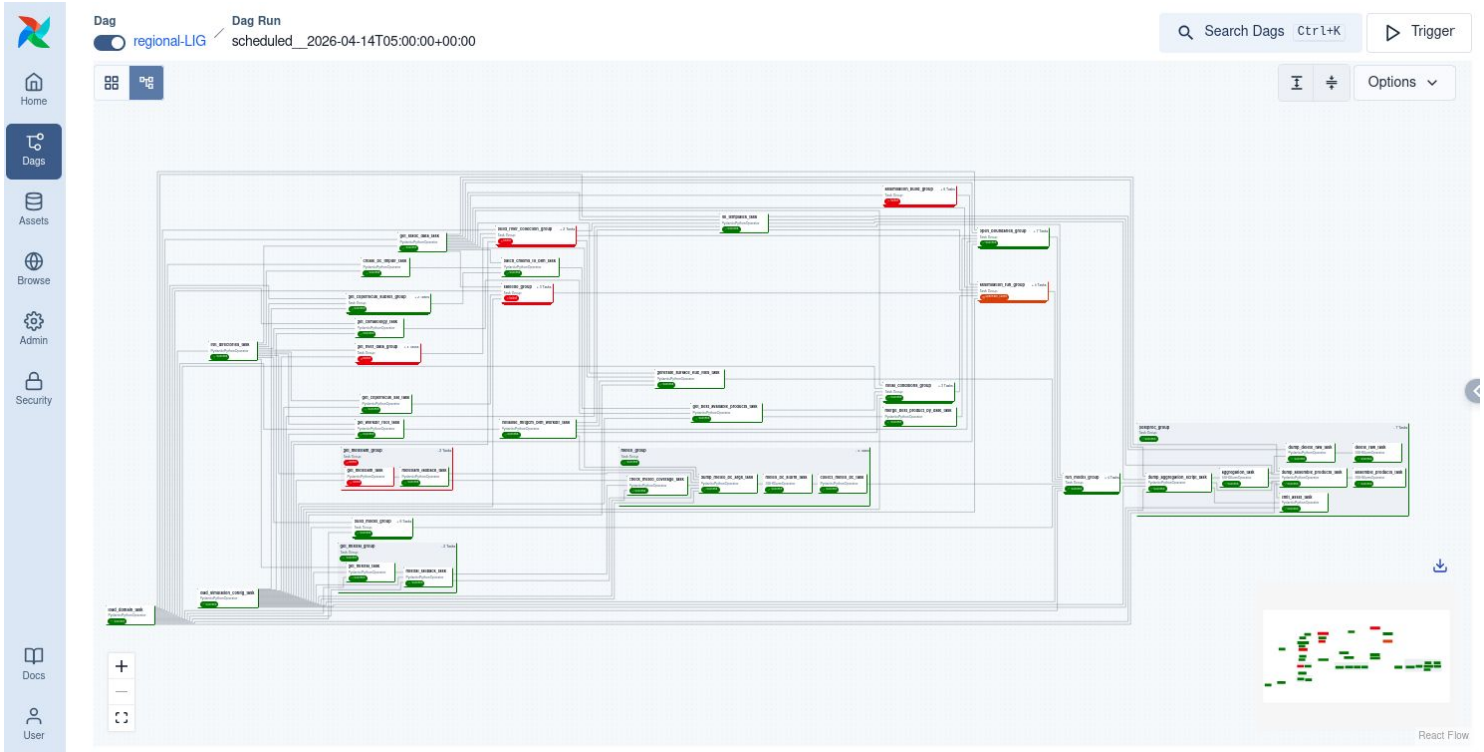
Workflows represented as direct acyclic graphs (**DAG**s) where the nodes are **tasks**, in the form of decorated Python functions.

Schedules and triggers DAGs.

Responsible for orchestrating workflows and fetching input data, monitoring and fallback.



# Apache Airflow for OO



# Airflow - HPC interaction

## Cluster interaction

A custom plugin connects Airflow to Slurm on Cineca Leonardo and G100 HPC clusters via SSH.

github.com/exactlab/airflow-slurm (GPL)

```python
from airflow_slurm.ssh_slurm_operator import SSHSlurmOperator


@dag(dag_id="slurm_job")
def slurm():
    slurm_task = SSHSlurmOperator(
        task_id='submit_slurm_task',
        ssh_conn_id='slurm', # Airflow connection ID (AIRFLOW_CONN_{C
        command='srun bash -c "sleep 20; echo Running task \$SLURM_PR
        slurm_options={
            "JOB_NAME": "example_job_name",
            "OUTPUT_FILE": "/path/to/slurmTEST-%j.out",
            "TIME": "01:00:00",
            "NODES": 2,
            "NTASKS": 8
        },
        tdelta_between_checks=10  # Poll interval (in seconds) for jo
    )
```

# HPC segment

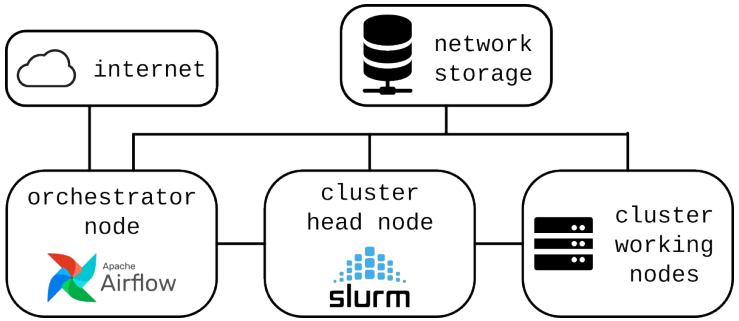
## Slurm workload manager

Our custom plugin reiles on Slurm need to know the status of jobs **reliably**.

## Polling needed

Slurm REST API is not available on Cineca Leonardo :(

Wishlist: webhook-like interface.

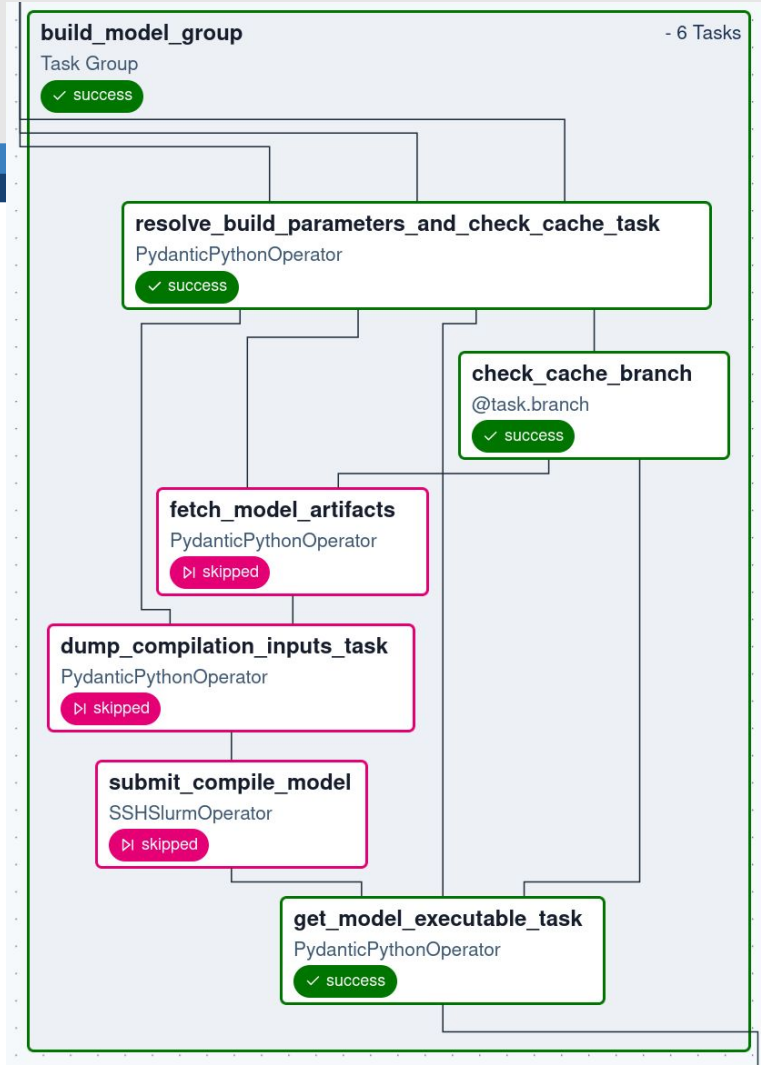


# Compiling for HPC

## Building...still harder than it should be

We use `modules` and `pkg-config` to ensure that all build dependencies for the model code be installed.

Still, automating building required 1000+ lines of custom Python to download the correct sources, patch them for each domain and build them.



# Compiling for HPC

## So hard that...

We containerised an HPC cluster to study the problem and test our infrastructure and released it as a FLOSS project.

codeberg.org/eXact-lab/vhpc (MIT)

# HPC segment

## I/O

Cineca Leonardo uses a Lustre parallel filesystem. MITgcm writes O($10^5$) tiny files as output of a typical run.


Aggregating them required careful optimisation and was made possible with the use of the `dask` library and `zarr` data format.


# Postprocessing optimisation

MITgcm

Fragmented output due to domain decomposition:

A few 100 TB split into 1-10 MB files




Zarr

A single zarr store with chunking automatically optimised for Lustre striding.

Enables efficient parallel downstream processing.

# Concluding remarks

# Conclusions

Lots of HPC work happens
**before**
the HPC process even starts!

Data-engineering is often a limiting factor in several HPC / AI workloads.

# Conclusions

## The Good

- Cloud-HPC architecture in a real world operational oceanography scenario
- Cloud systems are great at enriching HPC workflows with
  - Monitoring
  - Observability
  - Notifications
  - Fallback automation
- To be **open sourced in ~July!**

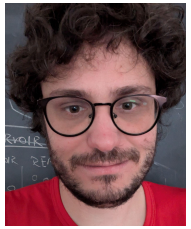
## The Bad

- Cloud-HPC interaction is still hacky
- Segmentation of networks and storage introduce artificial complexity
- Security is a serious issue: we need more technical skills and less bureaucracy.

## The Ugly

- It’s 2026 and we still can’t build software easily :(

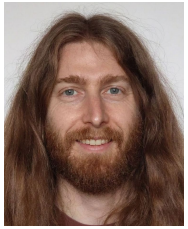
# The Team



Matteo Poggi, PhD
Physicist, co-technical lead

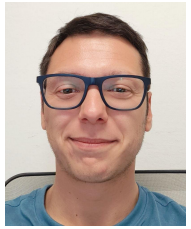
Alberto Pastorutti, PhD
Geologist, operations

Cecilia Zagni
Mathematical Engineer



Severino Zeni
Physicist

Yuri Chiucconi
Mathematician



Andrea Recchia
Physicist

Matteo Barnaba, PhD
Civil Engineer, operations

Funded by the European Union
NextGenerationEU

## In collaboration with:

OGS
Istituto Nazionale di Oceanografia e di Geofisica Sperimentale

CINECA

exact-lab.it

eXact lab
Solutions for the digital revolution

https://exact-lab.it
info@exact-lab.it