A HPDA-enabled environment for scalable climate data analysis

6th ENES HPC Workshop
Online
May 29, 2020

D. Elia[1,2], S. Fiore[1], C. Palazzo[1], A. D'Anca[1] and G. Aloisio[1,2]

[1] Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici (CMCC), Lecce, Italy
[2] University of Salento, Lecce, Italy

cmcc
Centro Euro-Mediterraneo sui Cambiamenti Climatici

CMCC Data Science Environment

✔ ***Main idea:*** *provide advanced data-science & learning capabilities, seamlessly integrated into a single high-performance problem solving environment to support climate change research at scale*

✔ **The goal:** enable climate scientists to address key scientific challenges and tackle much larger and complex science problems than those possible today in the climate change domain:

- manage large scientific **end-to-end climate experiments** (workflow support)
- perform **interactive data exploration** (e.g. Jupyter Notebooks)
- analyze **massive** datasets
- develop user-oriented **high-level data science applications**

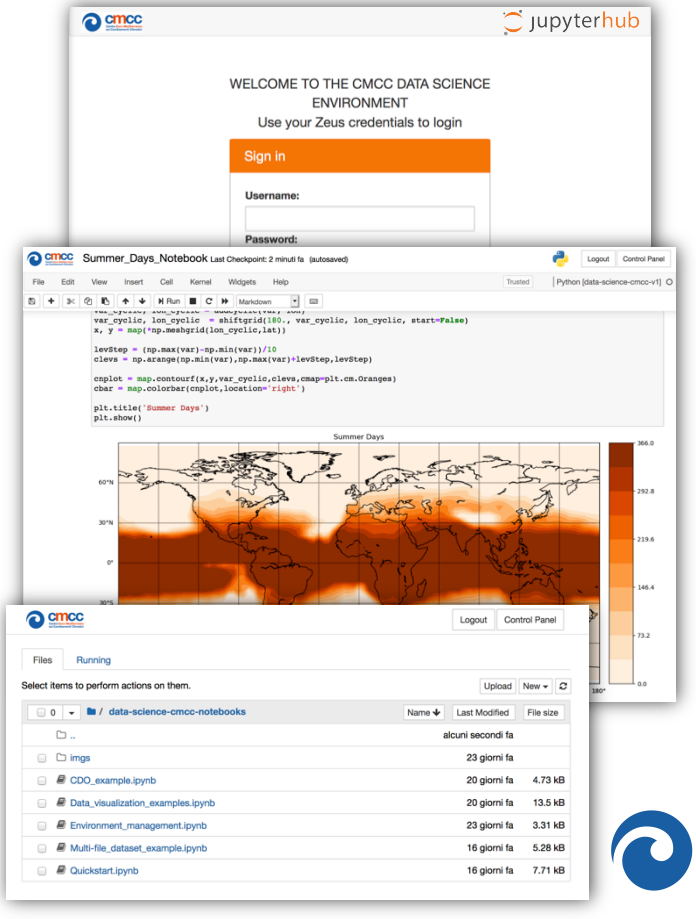
HPDA-enabled environment at CMCC

Infrastructure at CMCC SCC to host the environment software stack:

- ✔ *JupyterHub* & *Jupyter Notebooks* providing a graphical environment for user's experiments
- ✔ *Python modules* for data science, ML and visualization (e.g. NumPy, Pandas, Dask, Matplotlib, Cartopy, Keras)
- ✔ the *Ophidia HPDA framework*

Integration with *Zeus SuperComputer* infrastructure for transparent *compute and data resources access and user management*

Training notebooks for supporting users



Core services: the Ophidia framework

Ophidia (http://ophidia.cmcc.it) is a CMCC Foundation research project addressing data challenges for eScience

- ✔ A *High Performance Data Analytics* (HPDA) framework for multi-dimensional scientific data joining HPC paradigms with scientific data analytics approaches
- ✔ In-memory and *server-side data analysis* exploiting parallel computing techniques
- ✔ End-to-end mechanisms to support *interactive analysis*, *complex experiments* and *large workflows* on scientific datacubes
- ✔ Primarily exploited in climate change use cases

Ophidia

cmcc
Centro Euro-Mediterraneo
sui Cambiamenti Climatici

S. Fiore et al., "Ophidia: toward big data analytics for eScience", ICCS2013 Conference, Procedia Elsevier, 2013
S. Fiore et al., "Towards High Performance Data Analytics for Climate Change", ISC High Performance 2019, LNCS Springer, 2019

Ophidia 2.0 Architecture

Multi-interface interoperable front-end

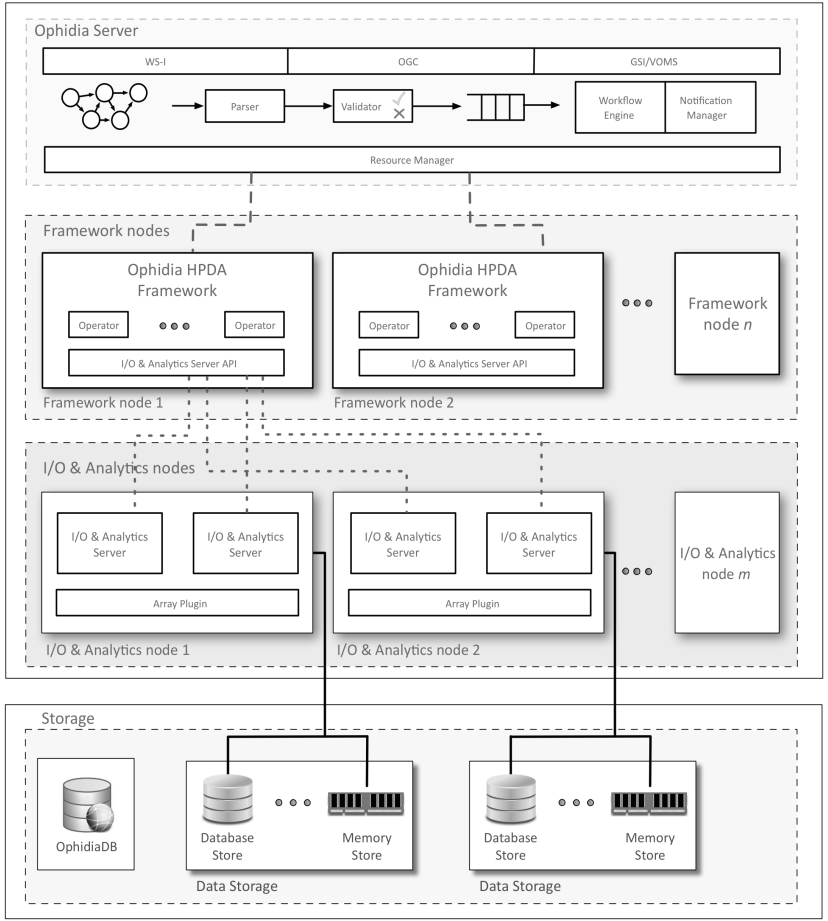
Modular and extensible software stack

Two-level runtime :

- ✔ Parallel framework
- ✔ I/O & analytics servers

Support for in-memory analytics

Data partitioned and distributed across the I/O & analytics nodes



On-demand instantiation of an Ophidia custer

Target environment: HPC cluster

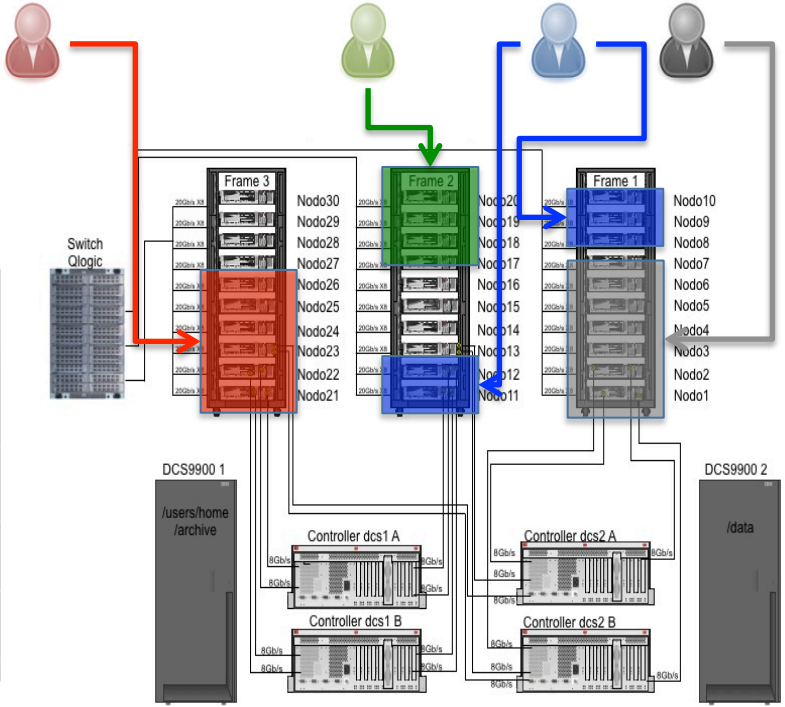
Deployment of I/O & analytics servers

```
oph_cluster action=deploy;nhost=64;cluster_name=new;

oph_cluster action=undeploy;cluster_name=new;
```

Zeus SuperComputer at CMCC: 1.2 PetaFlops, 348 nodes

Multiple isolated instances can be deployed simultaneously by different teams/users



Ophidia operators

CLASS	PROCESSING TYPE	OPERATOR(S)
I/O	Parallel	OPH_IMPORTNC, OPH_EXPORTNC, OPH_CONCATNC, OPH_RANDUCUBE
Time series processing	Parallel	OPH_APPLY
Datacube reduction	Parallel	OPH_REDUCE, OPH_REDUCE2, OPH_AGGREGATE
Datacube subsetting	Parallel	OPH_SUBSET
Datacube combination	Parallel	OPH_INTERCUBE, OPH_MERGECUBES
Datacube structure manipulation	Parallel	OPH_SPLIT, OPH_MERGE, OPH_ROLLUP, OPH_DRILLDOWN, OPH_PERMUTE
Datacube/file system management	Sequential	OPH_DELETE, OPH_FOLDER, OPH_FS
Metadata management	Sequential	OPH_METADATA, OPH_CUBEIO, OPH_CUBESCHEMA
Datacube exploration	Sequential	OPH_EXPLORECUBE, OPH_EXPLORENC

About 50 operators for data and metadata management

Array-based primitives

Data within datacubes fragments is physically stored in binary arrays

Ophidia provides a wide set of array-based primitives (around 100):

- ✔ Primitives come as plugins and are applied on a single datacube chunk (fragment)
- ✔ Some examples: predicates evaluation, statistical analysis, algebraic expression, regression, etc.

```
oph_apply query=oph_boxplot(oph_subarray(measure, 1, 8))
```

Single chunk or fragment (input)

INPUT TABLE 5 tuples x 50 elements										
ID	MEASURE									
1	10,73	8,66	7,83	11,20	6,02	1,95	9,25	16,11	...	8,70
2	22,85	17,84	21,82	18,57	14,81	18,71	19,31	19,83	...	21,13
3	19,89	30,17	24,95	30,07	25,40	26,31	22,95	23,18	...	24,82
4	11,60	12,49	13,91	13,53	9,48	15,27	13,05	14,17	...	11,66
5	13,94	12,43	17,95	14,70	20,41	14,46	15,37	18,00	...	18,30

Single chunk or fragment (output)

OUTPUT TABLE 5 tuples x 5 elements (summary)					
ID	MEASURE				
1	1,95	8,64	10,47	11,87	16,11
2	14,81	18,14	19,93	21,66	24,35
3	19,89	22,74	24,24	26,45	30,17
4	6,87	10,99	12,85	14,28	16,93
5	9,23	13,87	15,05	16,61	20,41

Programmatic support for data science applications

PyOphidia provides the Ophidia Python bindings for programmatic interaction with the framework and to retrieve/deserialize the results (e.g. in Jupyter Notebooks)

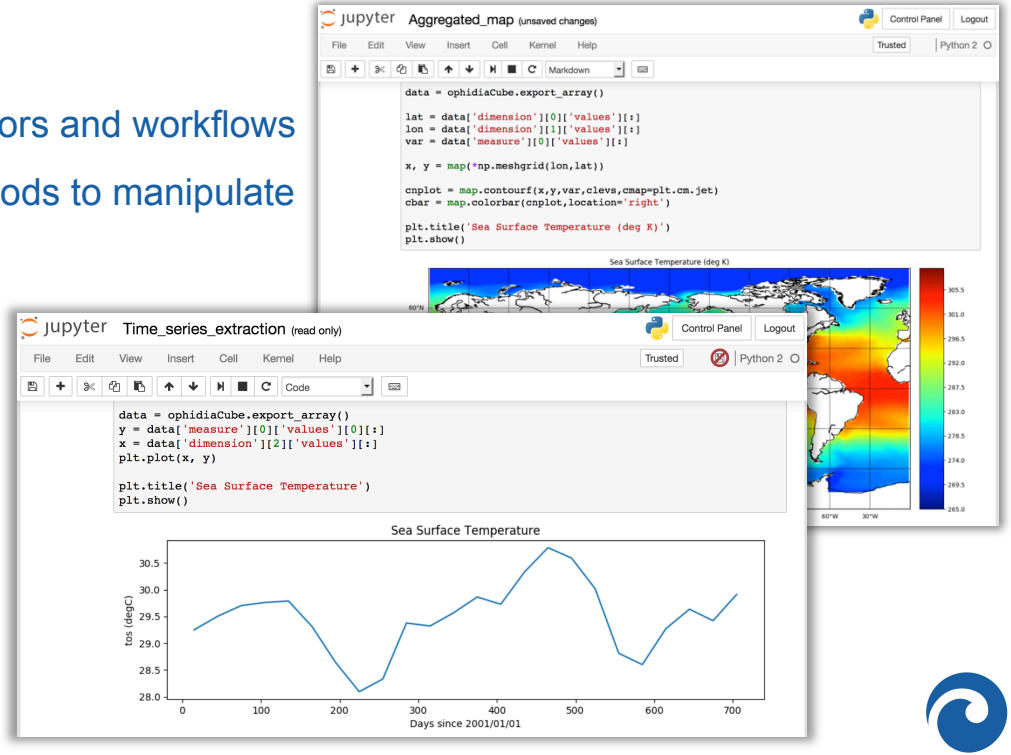
Two modules available:

- ✔ *Client class*: submissions of Ophidia operators and workflows
- ✔ *Cube class*: datacube abstraction and methods to manipulate and process cubes objects

```
from PyOphidia import cube, client
cube.Cube.setclient(read_env=True)

mycube =
cube.Cube.importnc(src_path='/public/data/ecas_training
/file.nc', measure='tos', imp_dim='time',
import_metadata='yes', ncores=5)
mycube2 = mycube.reduce(operation='max',ncores=5)
mycube3 = mycube2.rollup(ncores=5)
data = mycube3.export_array()

mycube3.exportnc2(output_path='/home/test',
export_metadata='yes')
```



https://pypi.org/project/PyOphidia/
https://anaconda.org/conda-forge/pyophidia

Python and HPC infrastructure transparency

PyOphidia class hides the HPC environment complexity

```
In [ ]: from PyOphidia import cube, client
        cube.Cube.setclient(read_env=True)

In [ ]: cube.Cube.cluster(action='deploy',host_partition='test_partition',nhost=4)

In [ ]: myCube = cube.Cube(src_path='/work/ophidia/tests/tasmax_day_CMCC-CESM_rcp85.nc',
            measure='tasmax', import_metadata='yes', imp_dim='time', description='Max Temps',
            nfrag=16, nhosts=4,
            host_partition='test2',
            ncores=2, nthreads=8
            )

In [ ]: myCube2 = maxtemp.apply(
            query="oph_predicate('oph_float','oph_int',measure,'x-298.15','>0','1','0')",
            ncores=2, nthreads=8
        )

In [ ]: myCube3 = myCube2.subset(subset_filter=1, subset_dims='time')

In [ ]: pythonData = myCube3.export_array(show_time='yes')

In [ ]: print(pythonData)

In [ ]: cube.Cube.cluster(action='undeploy',host_partition='test_partition')
```

Dynamic I/O & Analytics nodes allocation

Data partitioning and distribution

Framework operator parallelism

Ophidia-notebook data translation and transfer

I/O & Analytics nodes undeployment

Ophidia in ESiWACE2 project

Ophidia represents one of the applications/test case considered in the frame of the ESiWACE2 project (WP4 and WP5):

- ✔ One of the applications (HPDA) targeted by the ESDM PAV
  - Extensions for in-flight analytics are being developed
  - Some HPDA scientific use cases defined (preliminary implementation)
- ✔ Integration with the ESDM library for I/O over heterogeneous storage systems
- ✔ Benchmark in the context of PRACE resources for CoE

Ophidia

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

ESiWACE2 is a project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823988

Ophidia framework benchmark

Goal: benchmarking, tuning and optimization over a large-scale HPC machine of the Ophidia HPDA framework

Evaluate the performance of some Ophidia analytics test cases:

- ✔ multiple strong and weak scalability tests performed
- ✔ identify potential bottlenecks and baseline for comparison with future versions
- ✔ preliminary insight, technical report under preparation

Benchmark performed using the core hours awarded by PRACE (Call 18), in the context of the ESiWACE CoE, on MareNostrum 4 at the Barcelona Supercomputing Center (BSC)

The authors thankfully acknowledge the technical support provided by the Barcelona Supercomputing Center (BSC) and PRACE for awarding access to MareNostrum at BSC, Spain

Test cases evaluated

Test operations based on real-world use cases with (nested) primitives

✔ Using the *oph_apply* Ophidia operator

SHORT NAME	PRIMITIVES USED	TEST CASE DESCRIPTION
REGRESSION	1	Compute the time series trend with linear regression
SUMMER DAYS	3	Compute the number of days (on yearly basis) where the average temperature is above a given reference value*
SUBSET	2	Compute the average, std. deviation, minimum and maximum values from a subset of the original time series
DTR	4	Compute different statistics (average, variance, max, min, quartiles, etc.) on the whole time series of daily temperature variation
T90P	7	Compute the number of days (on yearly basis) where the average temperature is above the 90th percentile (evaluated on the whole time series)*

*Based in part on the ETCCDI climate indices: http://etccdi.pacificclimate.org/list_27_indices.shtml

Summary and future activities

Recap

- ✔ HPDA-enabled environment to support scientific data analysis activities
- ✔ Role of the Ophidia framework and its integration in the Python eco-system
- ✔ Preliminary experimental results concerning scalability up to a few thousand cores

Future activities

- ✔ Improve Ophidia targeting larger-scale HPDA scenarios
- ✔ Containerization of Ophidia over HPC infrastructures
- ✔ Other benchmarks of Ophidia targeting different scenarios
  - Comparison with the ESDM-PAV integrated version of Ophidia (ESiWACE2)

Thanks for your attention

These activities are supported in part by *ESiWACE2, a project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823988*

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

cmcc
Centro Euro-Mediterraneo sui Cambiamenti Climatici