The ENES Climate Analytics Service

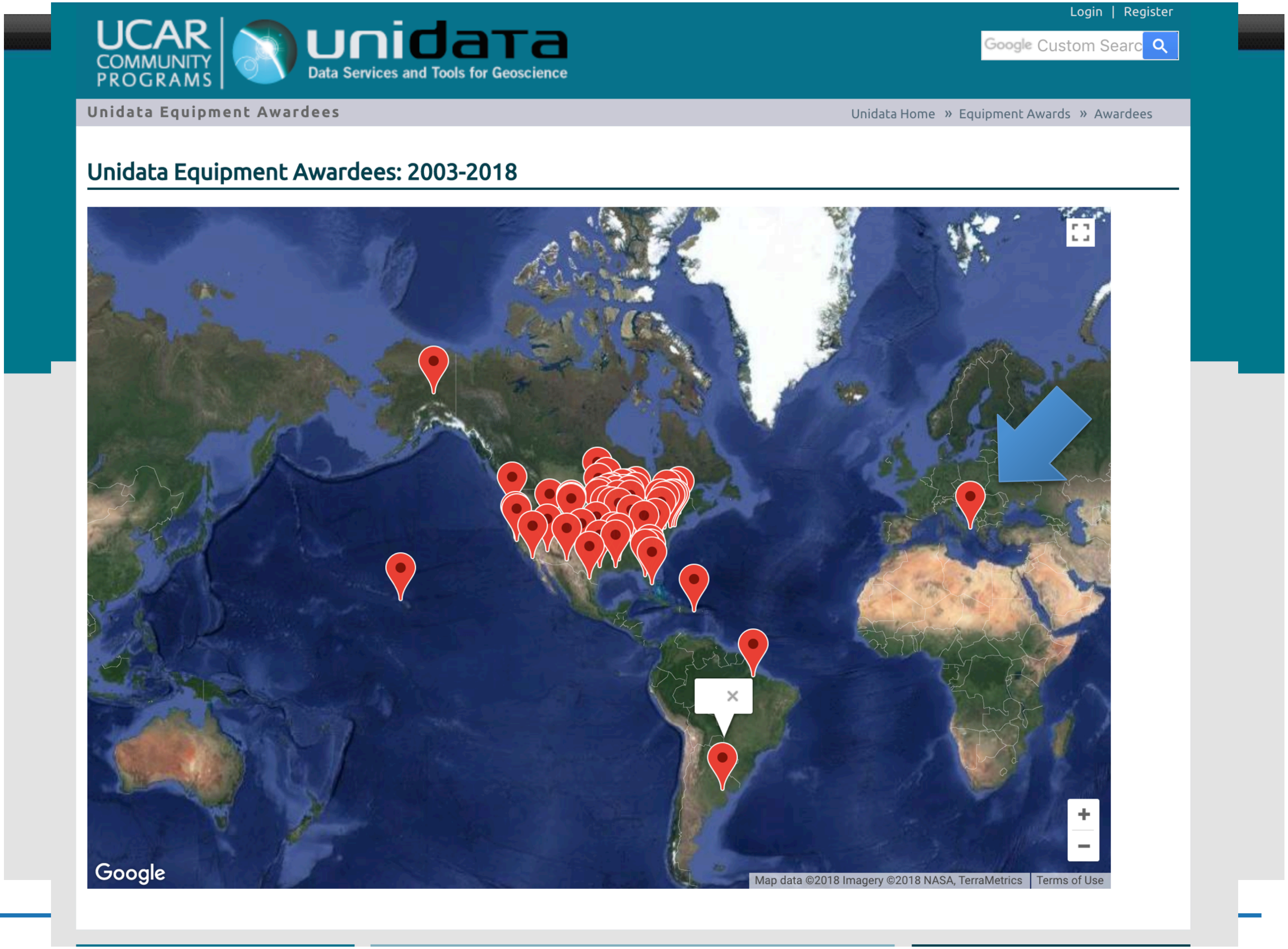
Sandro Fiore, Ph.D.
Advanced Scientific Computing Division

Prof. Giovanni Aloisio
CMCC Strategic Council & Director of the CMCC SCC

2018 UNIDATA User Workshop
Boulder, June 25th, 2018

EOSC-hub receives funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 777536.

UNIDATA Community Equipment Award 2011



Outline

- EOSC, ECAS and EOSC-hub
- Ophidia
  - Architecture 1.0
    - Storage model
    - Primitives
    - Data and metadata operators
  - Architecture 2.0
    - Workflow support
      - Some real use cases
    - PyOphidia
    - Native I/O server for in-memory analytics
- ECASLab in the context of EOSC-hub
    - Jupyter-Hub, Grafana, Workflow IDE
- Future work and conclusions
    - Looking forward
    - Website, github, youtube, pypi, …material for hands-on

EOSC,

ECAS & Ophidia

The context: European Open Science Cloud

- ✓ The **European Open Science Cloud (EOSC)** is an ambitious program will offer a **virtual environment** with **open** and **seamless services** for storage, management, **analysis** and **re-use of research data**, **across borders** and **scientifc disciplines** by federating existing scientifc data infrastructures, currently dispersed across disciplines and Member States.

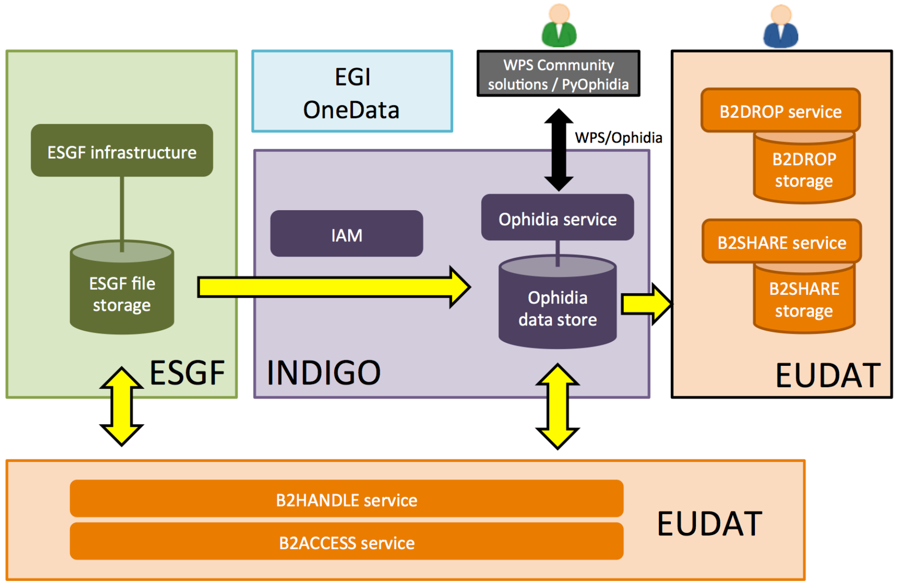
- ✓ This programme will deliver an **Open Data Science Environment** that **federates existing scientific data infrastructures** to offer European science and technology researchers and practitioners seamless access to services for storage, management, analysis and re-use of research data presently restricted by geographic borders and scientific disciplines.

ENES Climate Analytics Service (ECAS)

- ✓ The **ENES Climate Analytics Service (ECAS)**, proposed by CMCC & DKRZ in EOSC-hub supports climate data analysis
- ✓ It is one of the **EOSC-Hub Thematic Services** and has been ranked as the **1st out of 64** Thematic Service proposals
- ✓ ECAS builds on top of the **Ophidia big data analytics framework** with components from INDIGO-DataCloud, EUDAT and EGI
- ✓ The Analytics-Hub is a paradigm joining data and computing able to provide a **multi-model environment** for CMIP-based analytics experiments in ESGF

Ophidia

enes
EUROPEAN NETWORK FOR EARTH SYSTEM MODELLING



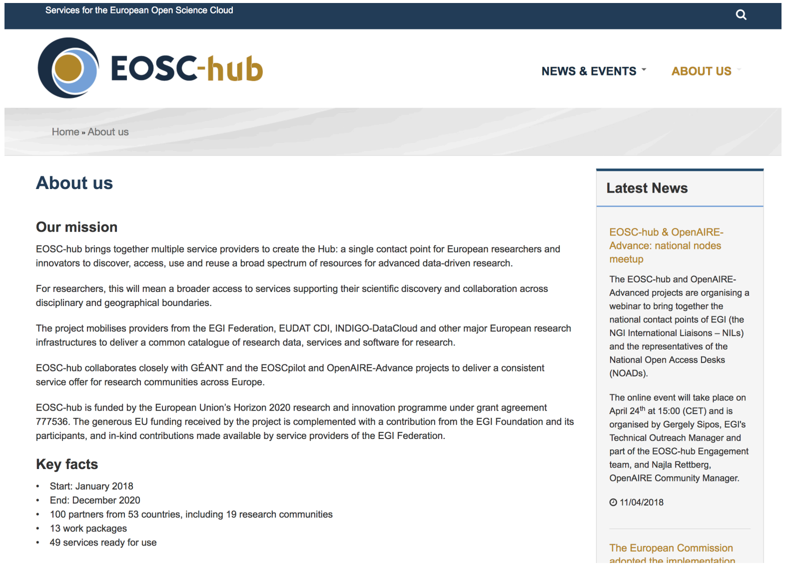
EOSC-hub

The European Commission launched the European Open ScienceCloud Initiative to capitalise on the data revolution. EOSC will provide European science, industry and public authorities with world-class digital infrastructure that bring state of the art computing and data storage capacity to the fingertips of any scientists and engineer in the EU.

EOSC-hub receives funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 777536.

ECAS and the European Open Science Cloud

- ECAS: a **data analytics service** for EOSC
  - **ENES**: European Network for Earth System Modelling
  - targets the climate community at large
- Involved institutions:
  - **DKRZ**: German Climate Computing Center
  - **CMCC**: Euro-Mediterranean Center on Climate Change Foundation
- Enable **server-side workflows** for Earth system researchers and beyond
- Induce cultural change: No more "**download and process at home**"
- **ECASLab** is the virtual environment for ECAS
  - Integrate several **UNIDATA** software (NetCDF lib, THREDDS and IDV)
- **ECAS is based on the Ophidia big data analytics framework**



EOSC-hub receives funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 777536.

Ophidia: a scientific big data analytics framework

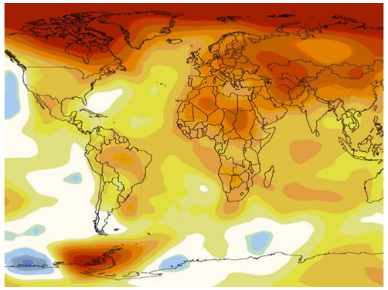
Ophidia (http://ophidia.cmcc.it) is a CMCC Foundation research project addressing fast and big data challenges for eScience

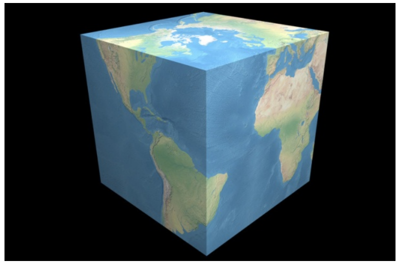
It provides support for declarative, parallel, server-side data analysis exploiting parallel computing techniques and database approaches

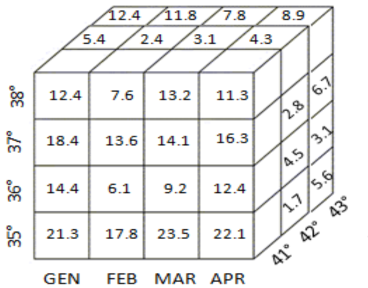
It provides end-to-end mechanisms to support complex experiments and large processing workflows on scientific datacubes

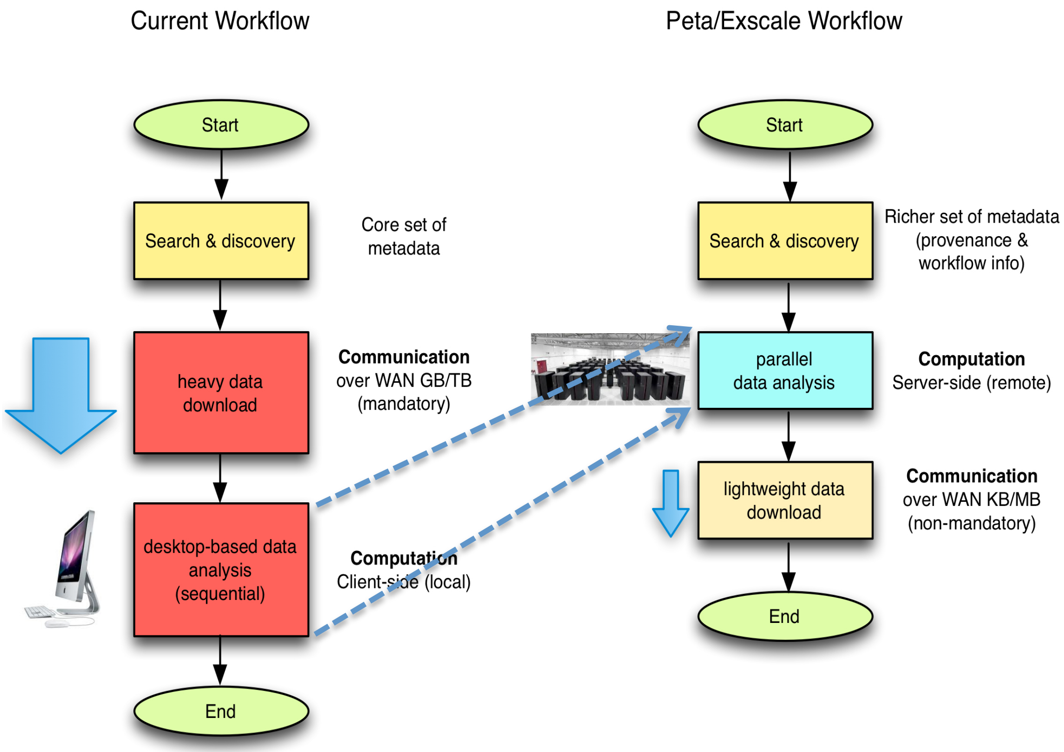
Big data challenges and the paradigm shift

Volume, variety, velocity are key challenges for big data in general and for climate change science in particular. Client-side, sequential and disk-based workflows are three limiting factors for the current scientific data analysis tools.









S. Fiore, A. D’Anca, C. Palazzo, I. Foster, D. N. Williams, G. Aloisio, “**Ophidia: toward bigdata analytics for eScience**”, ICCS2013 Conference, Procedia Elsevier, Barcelona, June 5-7, 2013

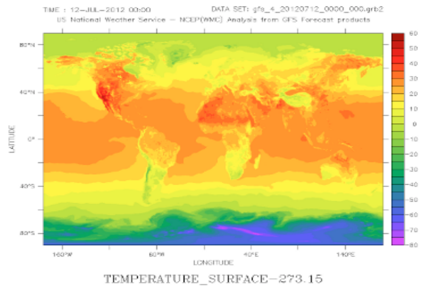
Data analytics requirements and use cases

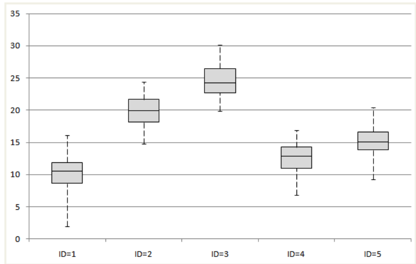
Requirements and needs focus on:

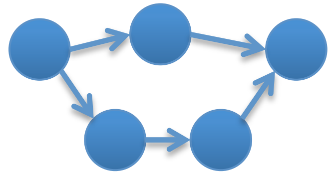
- *Time series analysis*
- *Data subsetting*
- *Model intercomparison*
- *Multimodel means*
- *Massive data reduction*
- *Data transformation (through array-based primitives)*
- *Param. Sweep experiments (same task applied on a set of data)*
- *Climate change signal*
- *Maps generation*
- *Ensemble analysis*
- *Data analytics worflow support*

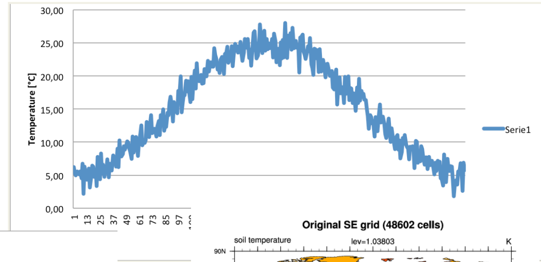
But also…

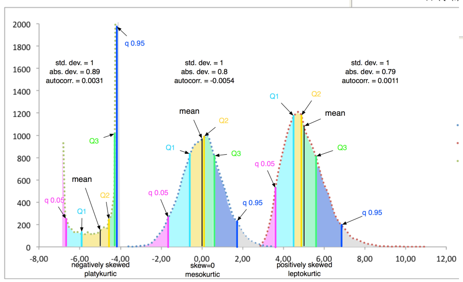
- *Performance*
- *re-usability*
- *extensibility*

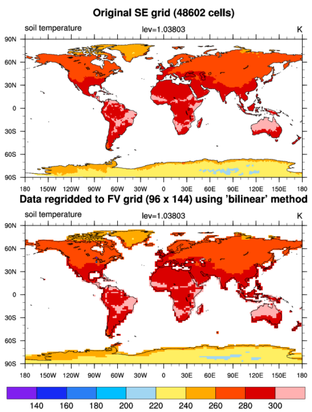








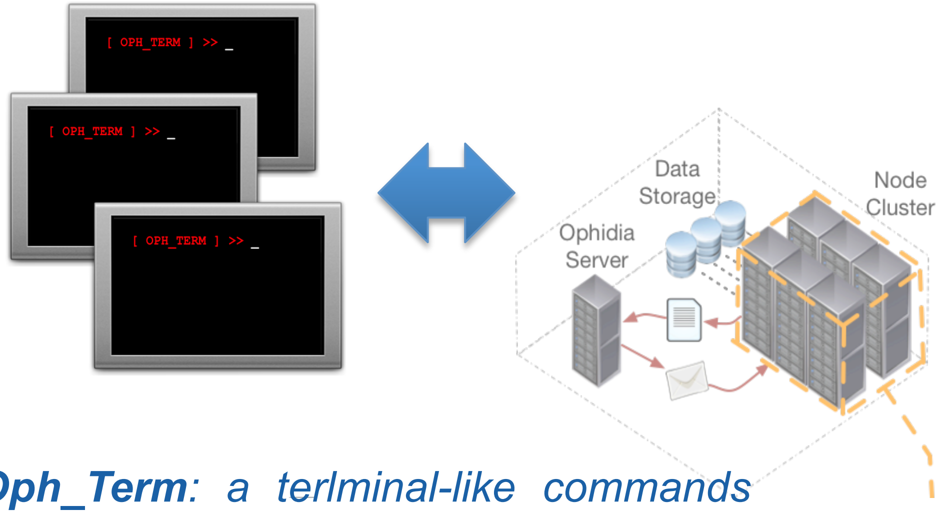


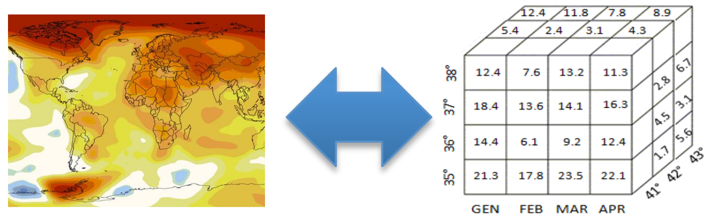


Ophidia in a nutshell

- ✔ ***Big data stack for scientific data analysis***
- ✔ ***Features:*** *time series analysis (array-based analysis), data subsetting (by value/index), data aggregation, model intercomparison, OLAP, etc.*
- ✔ *Use of parallel operators and parallel I/O*
- ✔ ***Support for complex workflows / operational chains***
- ✔ *Extensible:* ***simple API*** *to support framework extensions like new operators and array-based primitives*
  - ✔ *currently 50+ operators and 100+ primitives provided*
- ✔ ***Multiple interfaces*** *available (WS-I, GSI/VOMS, OGC-WPS).*
- ✔ *Programmatic access via C and* ***Python APIs***
- ✔ *Support for both* ***batch & interactive*** *data analysis*

Server-side paradigm and the datacube abstraction



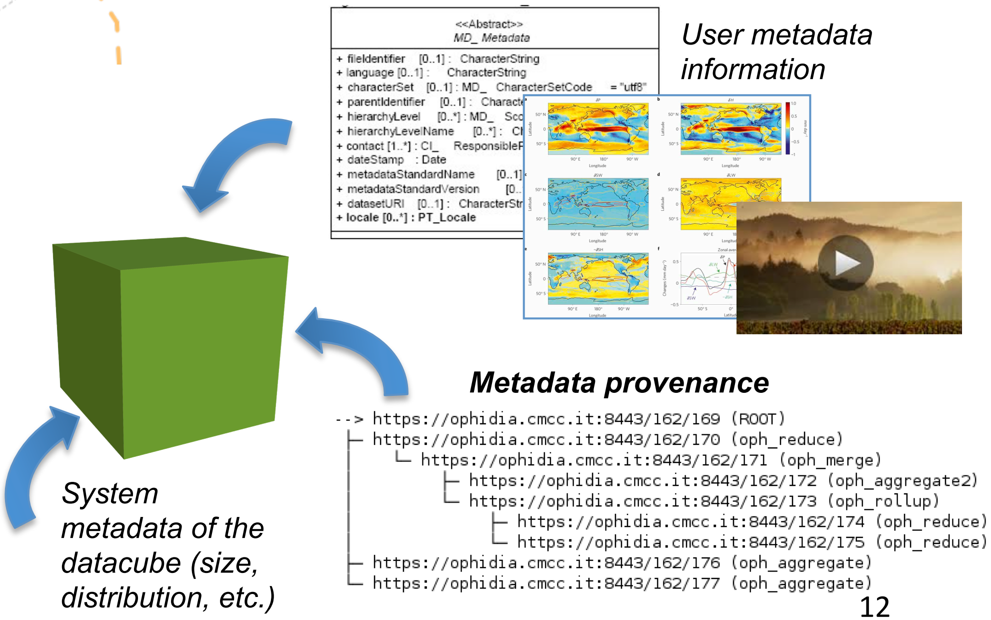


***Oph_Term**: a terlminal-like commands interpreter serving as a client for the Ophidia framework*

***Ophidia framework**: declarative, parallel server-side processing*

*Through the **oph_term** the user can send commands to the Ophidia framework to manipulate datasets*

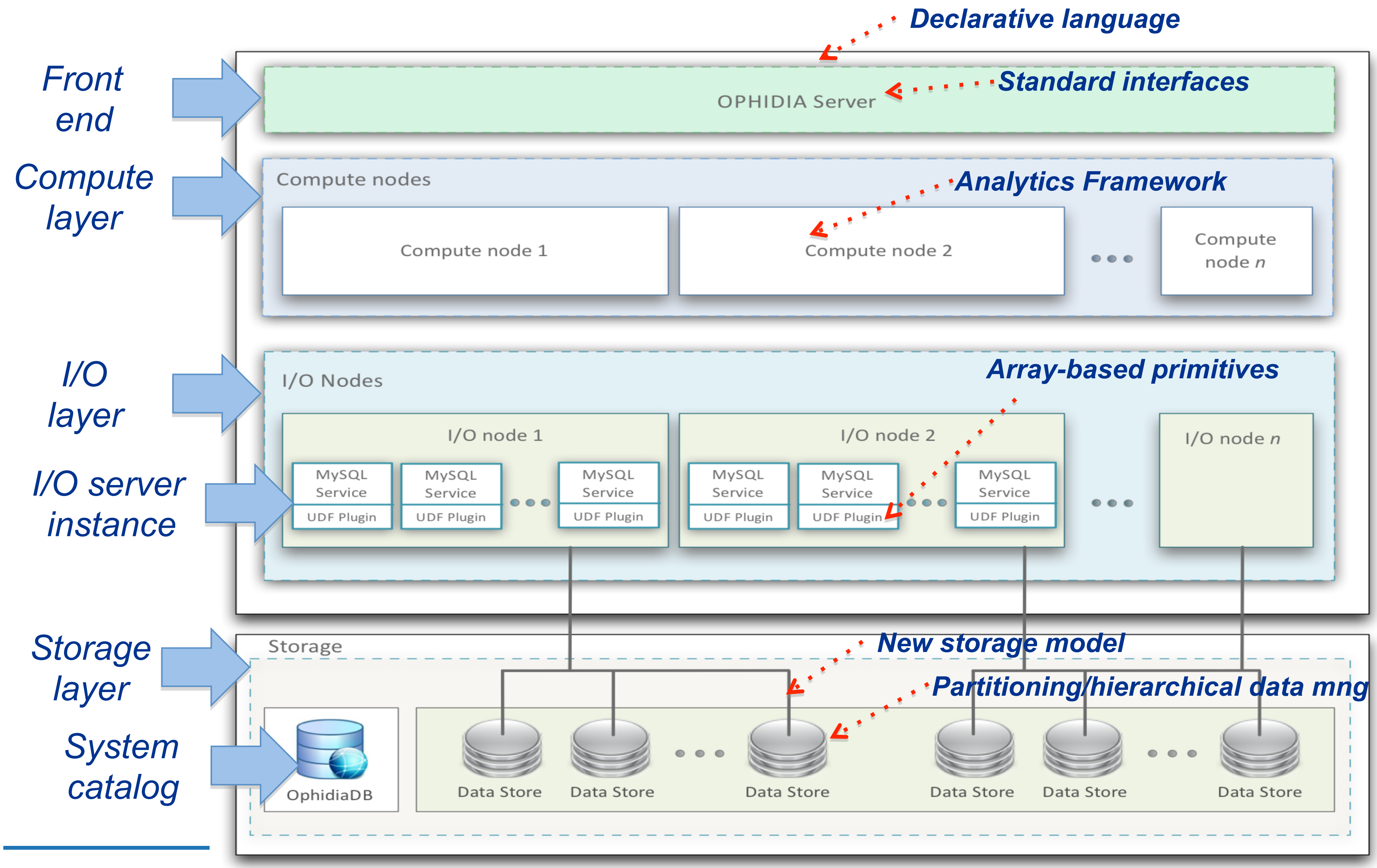
Three interaction modes:
Operators, Workflows, Python Apps



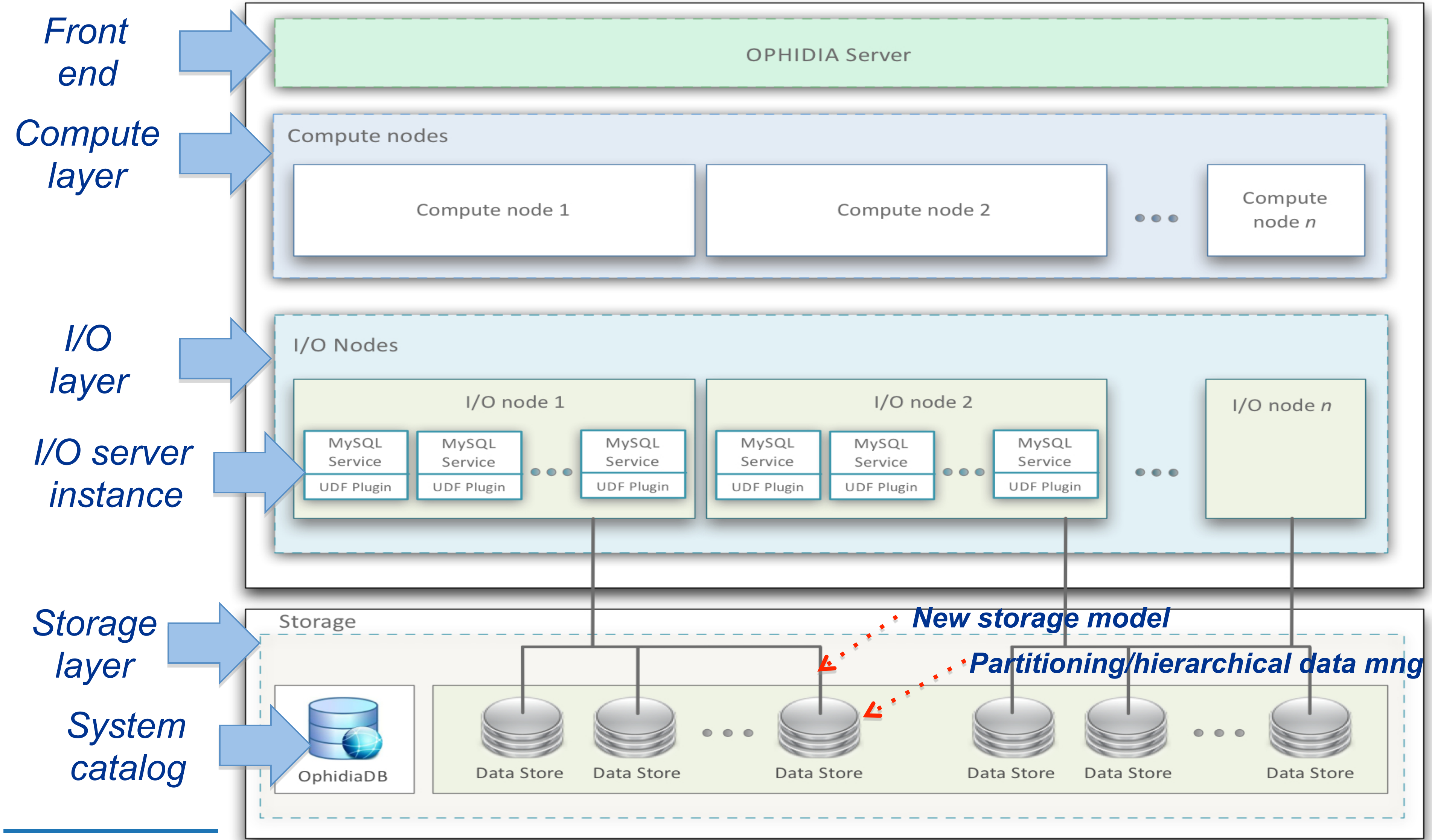
Ophidia architecture 1.0

Storage model, primitive & operators

Ophidia Architecture (sw stack view)



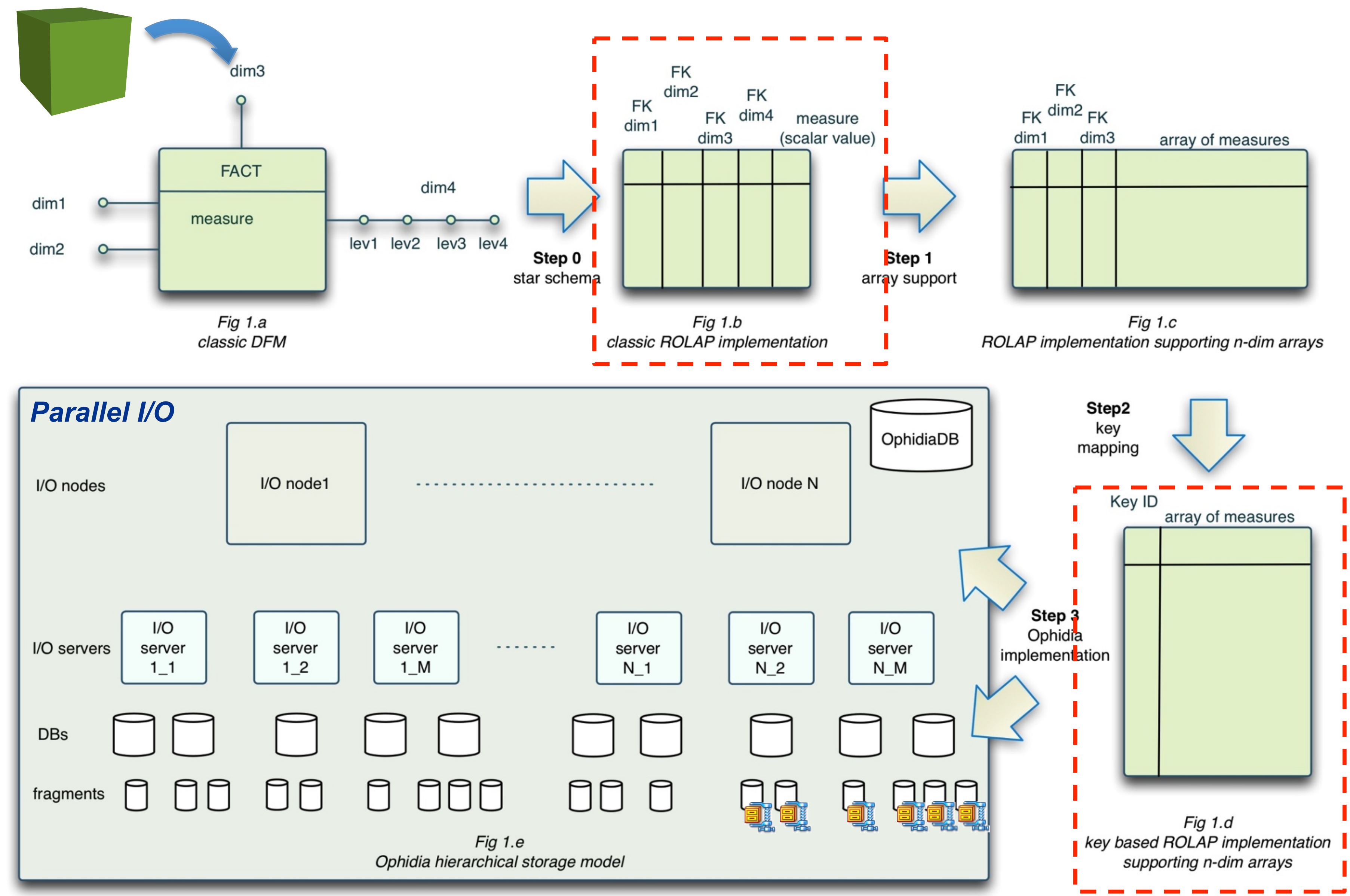
Storage model and chunks distribution



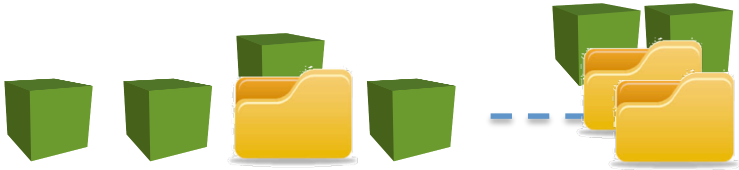
Ophidia storage model

- *The Ophidia storage model is a* ***two-step based evolution*** *of the* ***star schema*** *to support* ***scientific data management***
- *It relies on* ***implicit*** *(array-based) and* ***explicit*** *(tuple-based)* ***dimensions*** *for specific representations of data*
- *The first step includes the* ***support for array****-based data*
- *The second step includes a* ***key mapping*** *related to a set of foreign keys*
- *The second step makes the Ophidia storage model and implementation* ***independent of the number of dimensions****!*

Storage model (dimension-independent) & implementation
Array-based support and hierarchical storage

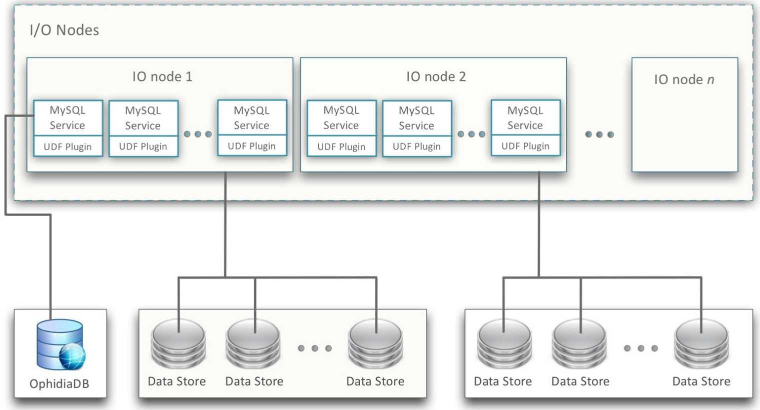


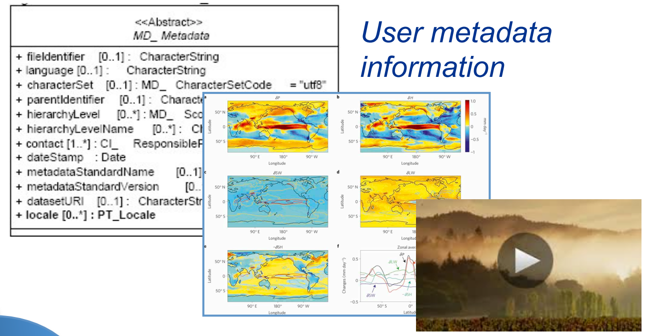
Data abstraction: cube space perspective



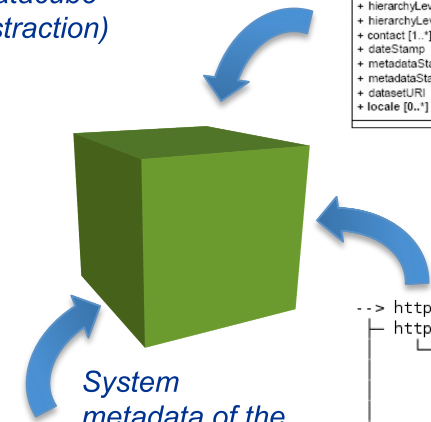
User perspective (datacube abstraction)

System perspective (internal storage representation)





User metadata information



System metadata of the datacube (size, distribution, etc.)

Metadata provenance

```
--> https://ophidia.cmcc.it:8443/162/169 (ROOT)
 ├ https://ophidia.cmcc.it:8443/162/170 (oph_reduce)
 │   └ https://ophidia.cmcc.it:8443/162/171 (oph_merge)
 │       ├ https://ophidia.cmcc.it:8443/162/172 (oph_aggregate2)
 │       └ https://ophidia.cmcc.it:8443/162/173 (oph_rollup)
 │           ├ https://ophidia.cmcc.it:8443/162/174 (oph_reduce)
 │           └ https://ophidia.cmcc.it:8443/162/175 (oph_reduce)
 ├ https://ophidia.cmcc.it:8443/162/176 (oph_aggregate)
 └ https://ophidia.cmcc.it:8443/162/177 (oph_aggregate)
```

Manage the Ophidia file system

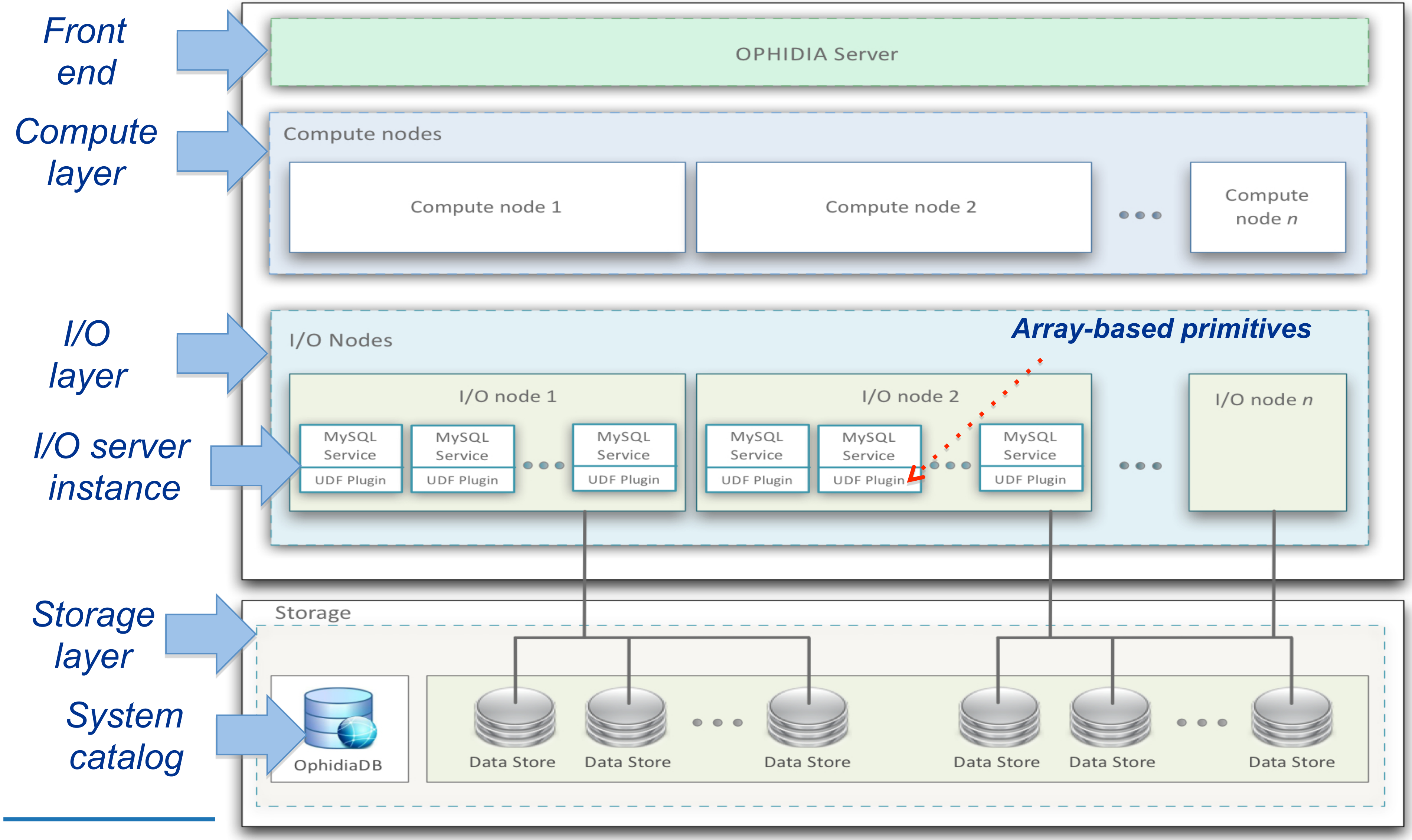
CMD	BEHAVIOR
cd	change directory
mkdir	create a new folder
rm	remove an empty folder or hide (logically delete) a container
ls	list subfolders and containers in a folder
mv	move/rename a folder or a container
…	…

Metadata associated to the datacubes

TYPE	CONTENT
Text	Plain text metadata
image	Binary string representation of an image
video	Binary string representation of a video
audio	Binary string representation of an audio stream
url	Text representing an URL

Search & Discovery

Array-based primitives



Array based primitives (about 100)

- *Ophidia provides a **wide set of array-based primitives** to perform data summarization, sub-setting, predicates evaluation, statistical analysis, compression, etc.*
- *Primitives come as plugins and are applied on a single datacube chunk (fragment)*
- *They are provided both for **byte**-oriented and **bit**-oriented arrays*
- ***Primitives can be nested** to get more complex functionalities*
- ***Compression is a primitive too!***
- *New primitives can be easily integrated as additional plugins*

Array based primitives: OPH_MATH ("SIGN")

oph_math(measure, "OPH_SIGN", "OPH_DOUBLE")

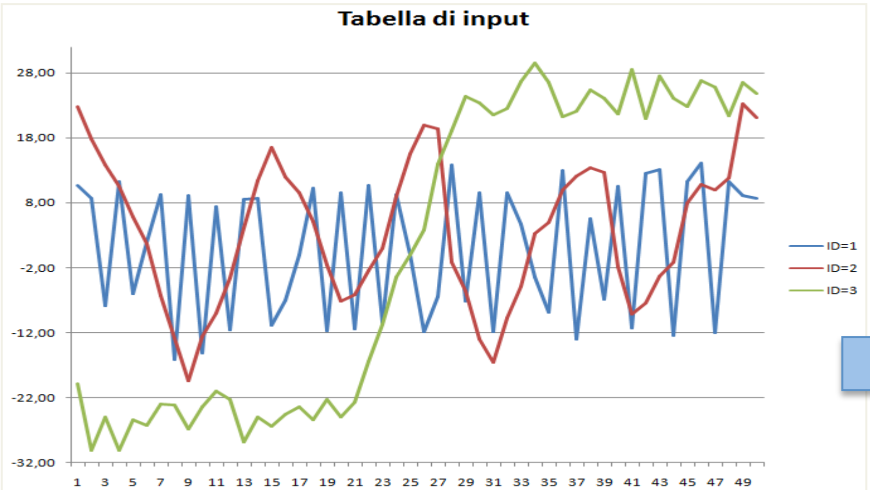
oph_math(measure,"OPH_MATH_SIGN", "OPH_DOUBLE")

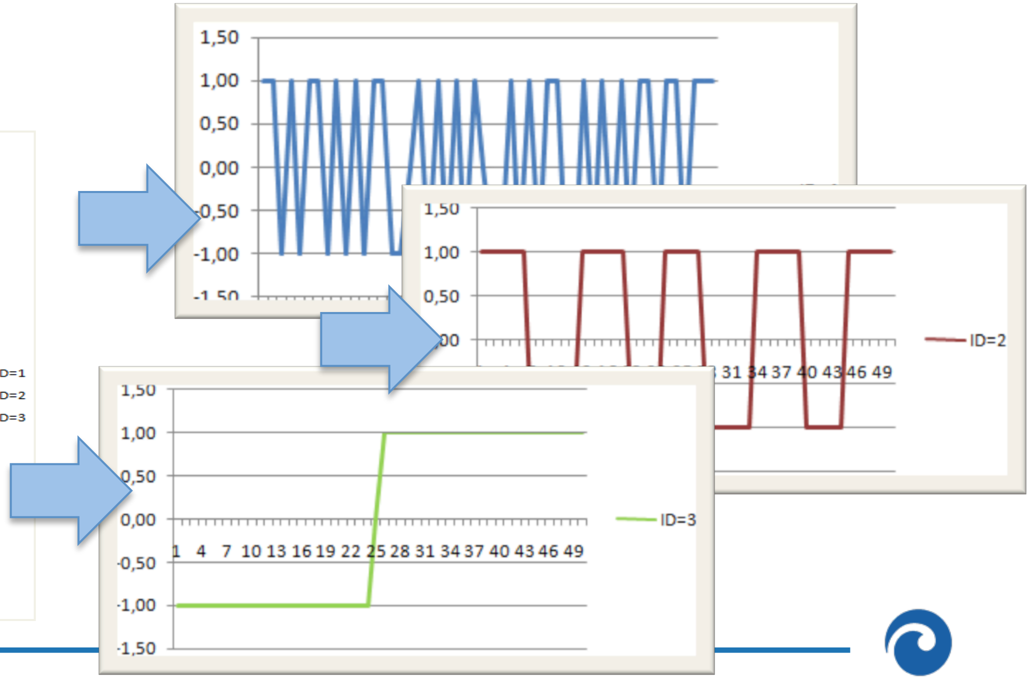
TABELLA INPUT 3 x 50								
ID	**MEASURE**							
1	10,73	8,66	-7,83	11,2	-6,02	1,95	...	8,70
2	22,85	17,84	13,82	10,57	5,81	1,71	...	21,13
3	-19,89	-30,17	-24,95	-30,07	-25,4	-26,31	...	24,82

Single chunk or fragment (input)

TABELLA OUTPUT 3 x 50								
ID	**MEASURE**							
1	1	1	-1	1	-1	1	...	1
2	1	1	1	1	1	1	...	1
3	-1	-1	-1	-1	-1	-1	...	1

Single chunk or fragment (output)





Array-based primitives: OPH_MATH support

*oph_math(measure, **OPH_MATH_FUNCTION**, "OPH_DOUBLE")*

SQL query

Ophidia math plugin

Ophidia typing

OPH_MATH_FUNCTION can be one of the macros in the table below

OPH_MATH_FUNCTION MACROS

OPH_MATH_ABS	OPH_MATH_DEGREES	OPH_MATH_RAND
OPH_MATH_ACOS	OPH_MATH_EXP	OPH_MATH_ROUND
OPH_MATH_ASIN	OPH_MATH_FLOOR	**OPH_MATH_SIN**
OPH_MATH_ATAN	OPH_MATH_LN	OPH_MATH_SIGN
OPH_MATH_CEIL	OPH_MATH_LOG10	OPH_MATH_SQRT
OPH_MATH_COS	OPH_MATH_LOG2	OPH_MATH_TAN
OPH_MATH_COT	OPH_MATH_RADIANS	…

Array based primitives: OPH_BOXPLOT

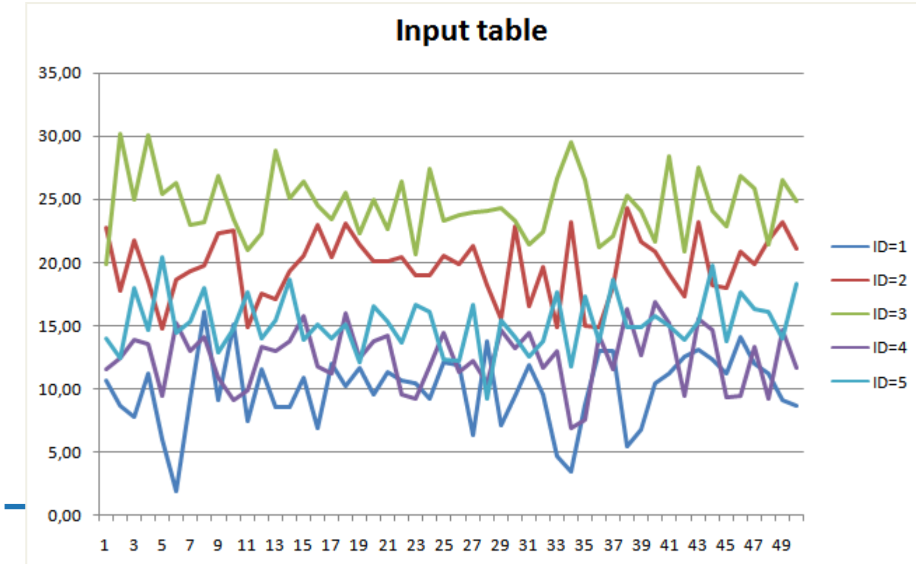
oph_boxplot*(measure, "OPH_DOUBLE")*

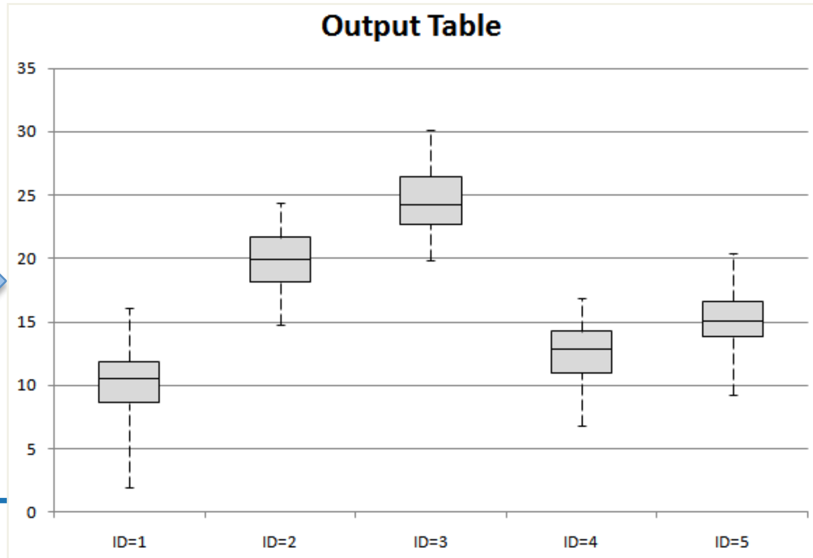
Single chunk or fragment (input)

INPUTTABLE 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)

OUTPUTTABLE 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





Array based primitives: nesting feature

oph_boxplot(oph_subarray(oph_uncompress(measure), 1,18), "OPH_DOUBLE")

Single chunk or fragment (input)

INPUTTABLE 5 tuples x 50 elements

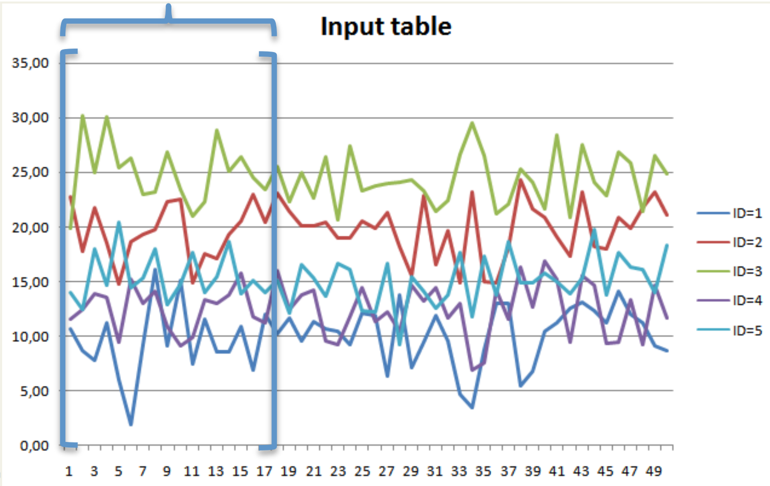
ID	MEASURE									
1	10,73	8,66	7,83	11,20	6,02	1,95	...	16,11	...	8,70
2	22,85	17,84	21,82	18,57	14,81	18,71	...	19,83	...	21,13
3	19,89	30,17	24,95	30,07	25,40	26,31	...	23,18	...	24,82
4	11,60	12,49	13,91	13,53	9,48	15,27	...	14,17	...	11,66
5	13,94	12,43	17,95	14,70	20,41	14,46	...	18,00	...	18,30

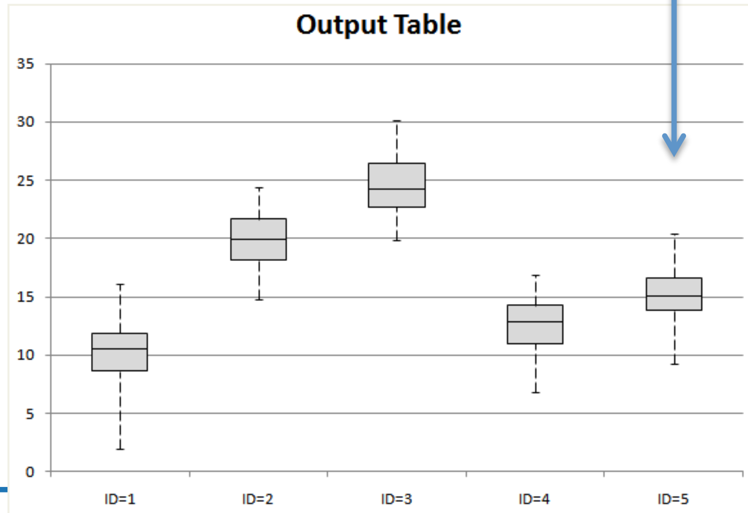
Single chunk or fragment (output)

OUTPUTTABLE 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

subarray(measure, 1,18)





Array based primitives: oph_aggregate

oph_aggregate(measure,"oph_avg")

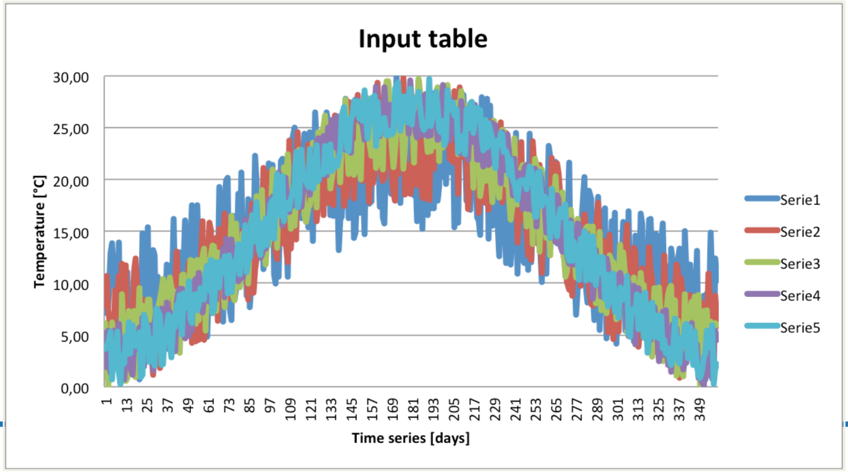
Single chunk or fragment (input)

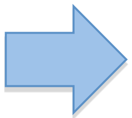
INPUT TABLE 5 tuples x 360 elements										
ID	**MEASURE**									
1	8,40	7,73	7,36	12,68	13,34	11,17	9,09	2,04	...	7,75
2	7,85	10,71	7,23	5,14	4,68	2,61	9,17	8,50	...	6,57
3	6,40	3,48	0,44	2,81	6,16	2,01	3,61	3,83	...	5,88
4	5,60	4,68	5,54	5,84	5,47	5,37	5,30	7,24	...	3,06
5	3,55	4,10	4,59	5,07	6,97	2,07	3,06	3,06	...	7,88

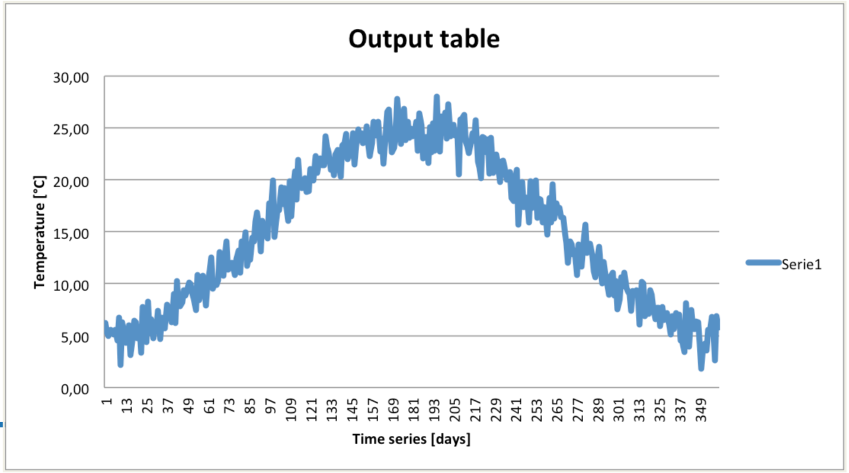
Vertical aggregation

OUTPUT TABLE 1 tuple x 360 elements							
ID	**MEASURE**						
1	6,25	5,35	5,00	5,57	5,41	...	5,11

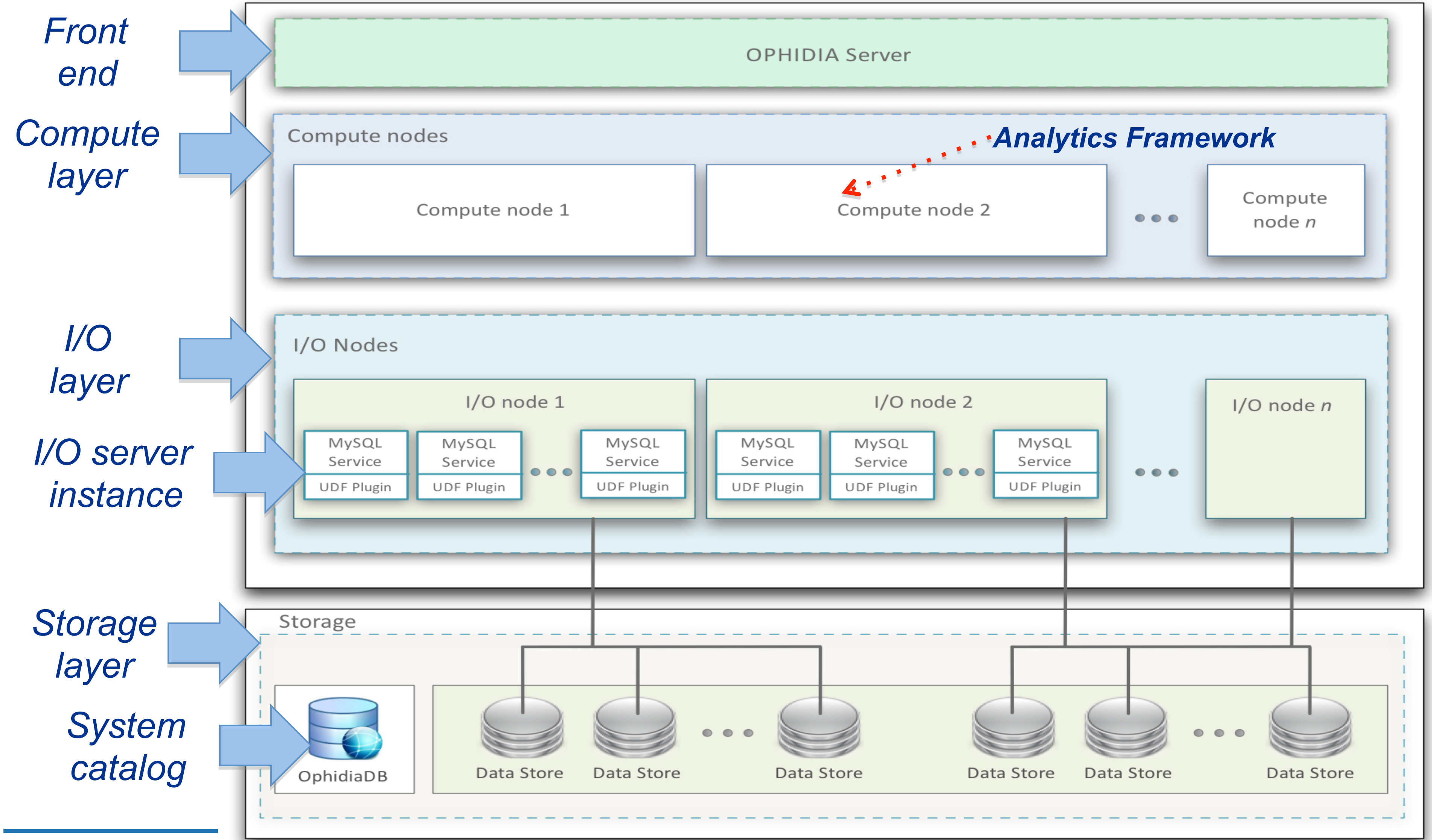
Single chunk or fragment (output)







Analytics framework and operators



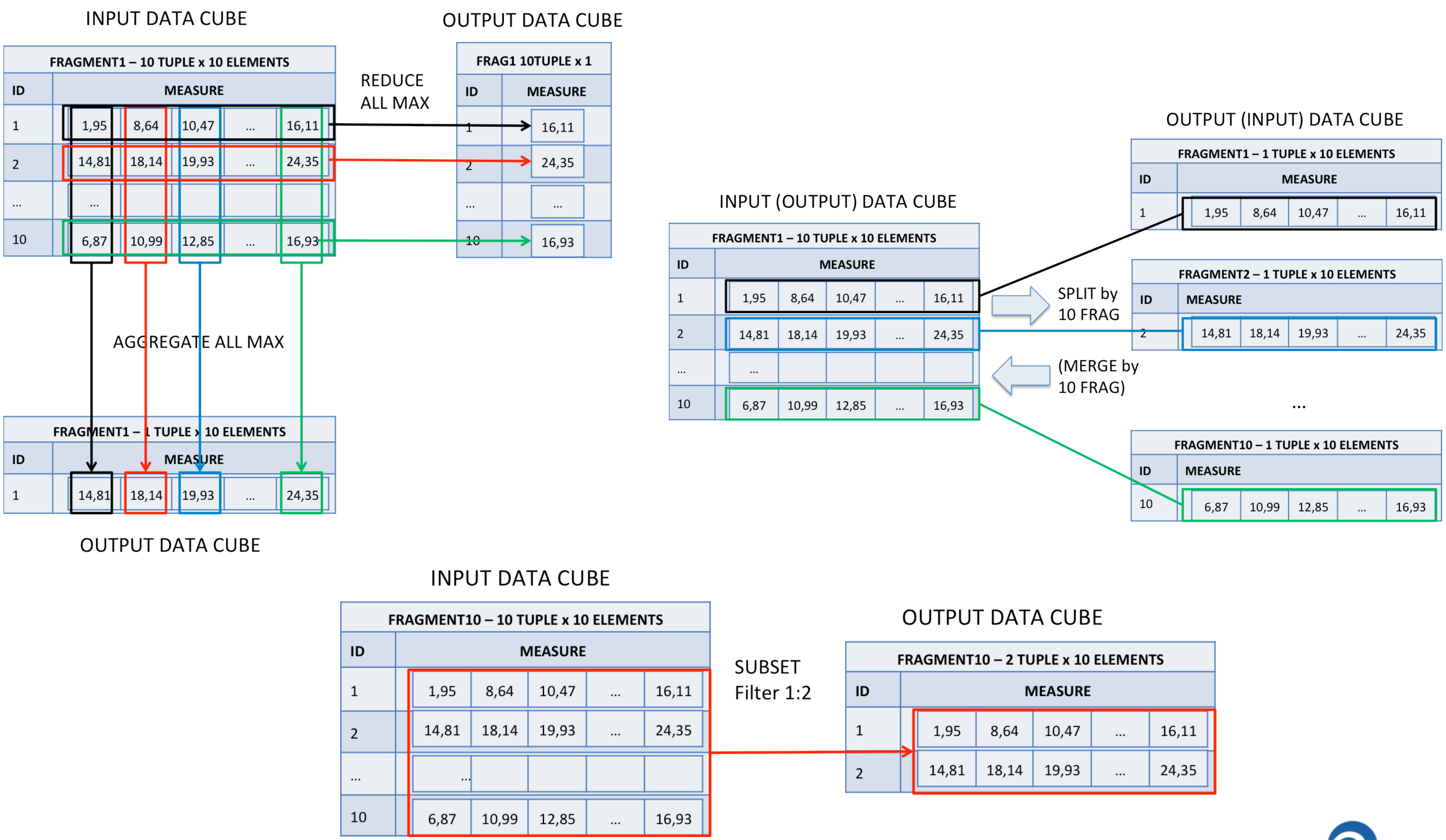
The analytics framework: datacube operators

Data Operator	Description
OPH_CONCATNC	Concatenates a NetCDF file to a data cube.
OPH_DELETE	Deletes a data cube.
OPH_DUPLICATE	Duplicates a data cube.
OPH_EXPLORECUBE	Shows the content of a data cube.
OPH_EXPORTNC	Exports a whole data cube into a single NetCDF file.
OPH_IMPORTNC	Creates new a data cube importing data from a NetCDF file.
OPH_INTERCOMPARISON	Generates the difference value-by-value between two homogeneous data cubes.
OPH_INTERCUBE	It executes an operation between two data cubes and returns a new data cube as result of the specified operation applied element by element.
OPH_MERGECUBES	Merges the measures of n input data cubes creating a new data cube with the union of the n measures.
OPH_PUBLISH	Generates web pages representing the data stored in the fragments.
OPH_RANDCUBE	Creates a new data cube with random data.
OPH_REDUCE	Applies a data reduction operation along one or more implicit dimensions.
OPH_SCRIPT	Executes a bash script.
OPH_SUBSET	Extracts a subset from a data cube using the values of the dimensions.

Metadata Operator	Description
OPH_CUBEELEMENTS	Computes and displays the total number of elements contained in a data cube.
OPH_CUBEIO	Shows the provenance of a data cube.
OPH_CUBESCHEMA	Displays the metadata and dimension information associated to a data cube.
OPH_CUBESIZE	Computes and displays the total size (on disk) of a data cube.
OPH_FIND	Finds a data cube.
OPH_LIST	Displays the list of data cubes and containers available.
OPH_LOGGINGBK	Shows session and job information.
OPH_MAN	Shows a description about an operator or primitive.
OPH_METADATA	Manages metadata information.
OPH_OPERATORS_LIST	Displays the list of available operators.

About 50 operators for data and metadata processing

The analytics framework: “datacube” operators



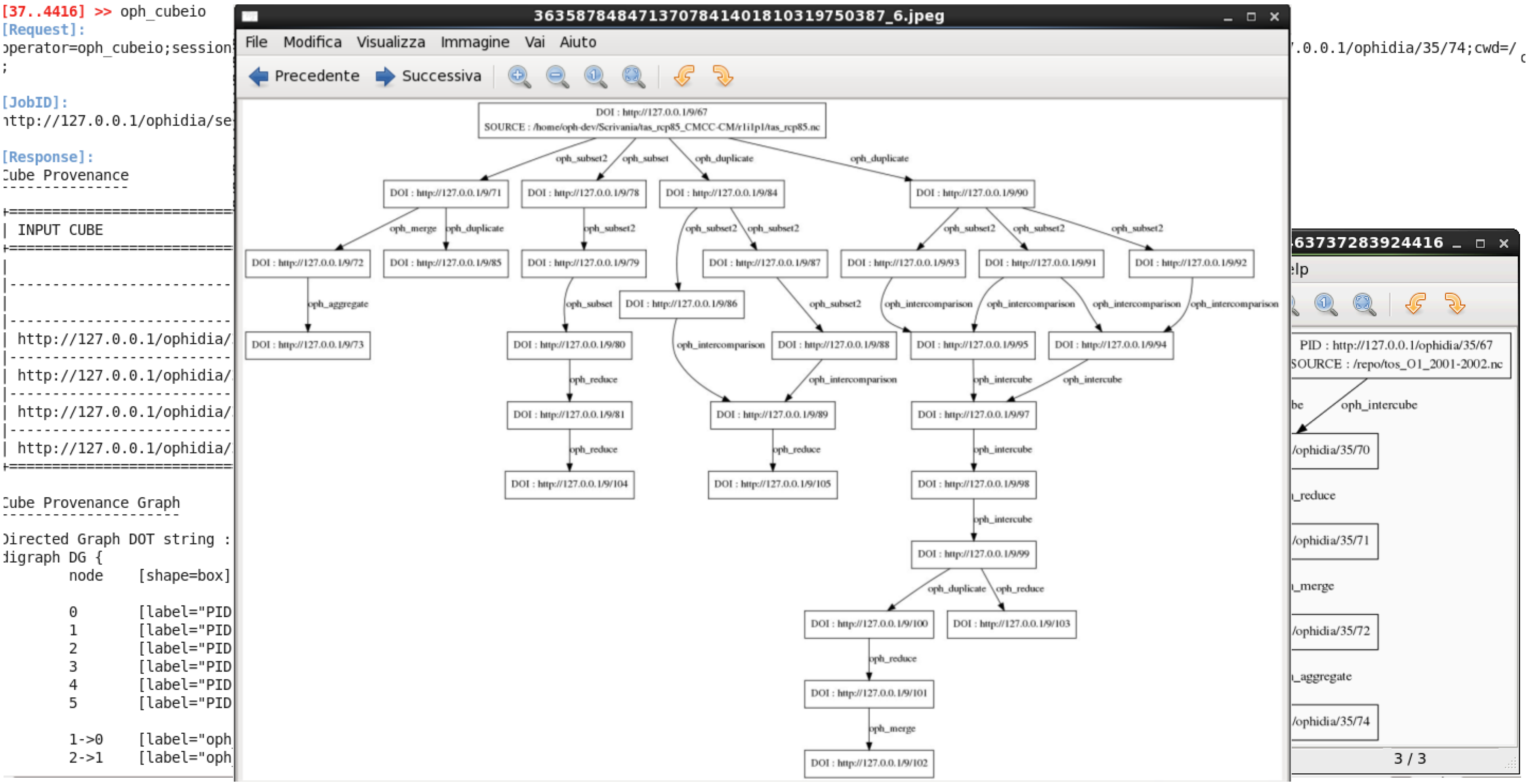
The analytics framework: “data” operators

```
[37..4416] >> oph_explorecube cube=http://127.0.0.1/ophidia/35/67;subset_dims=lat|lon|time;subset_filter=39:42|15:19|1:275;show_time=yes;
[Request]:
operator=oph_explorecube;cube=http://127.0.0.1/ophidia/35/67;subset_dims=lat|lon|time;subset_filter=39:42|15:19|1:275;show_time=yes;sessionid=http://127.0.0.1/ophidia/sessions/3
74383780832141666641463737283924416/experiment;exec_mode=sync;ncores=1;cwd=/;

[JobID]:
http://127.0.0.1/ophidia/sessions/374383780832141666641463737283924416/experiment?106#224

[Response]:
tos
---
```

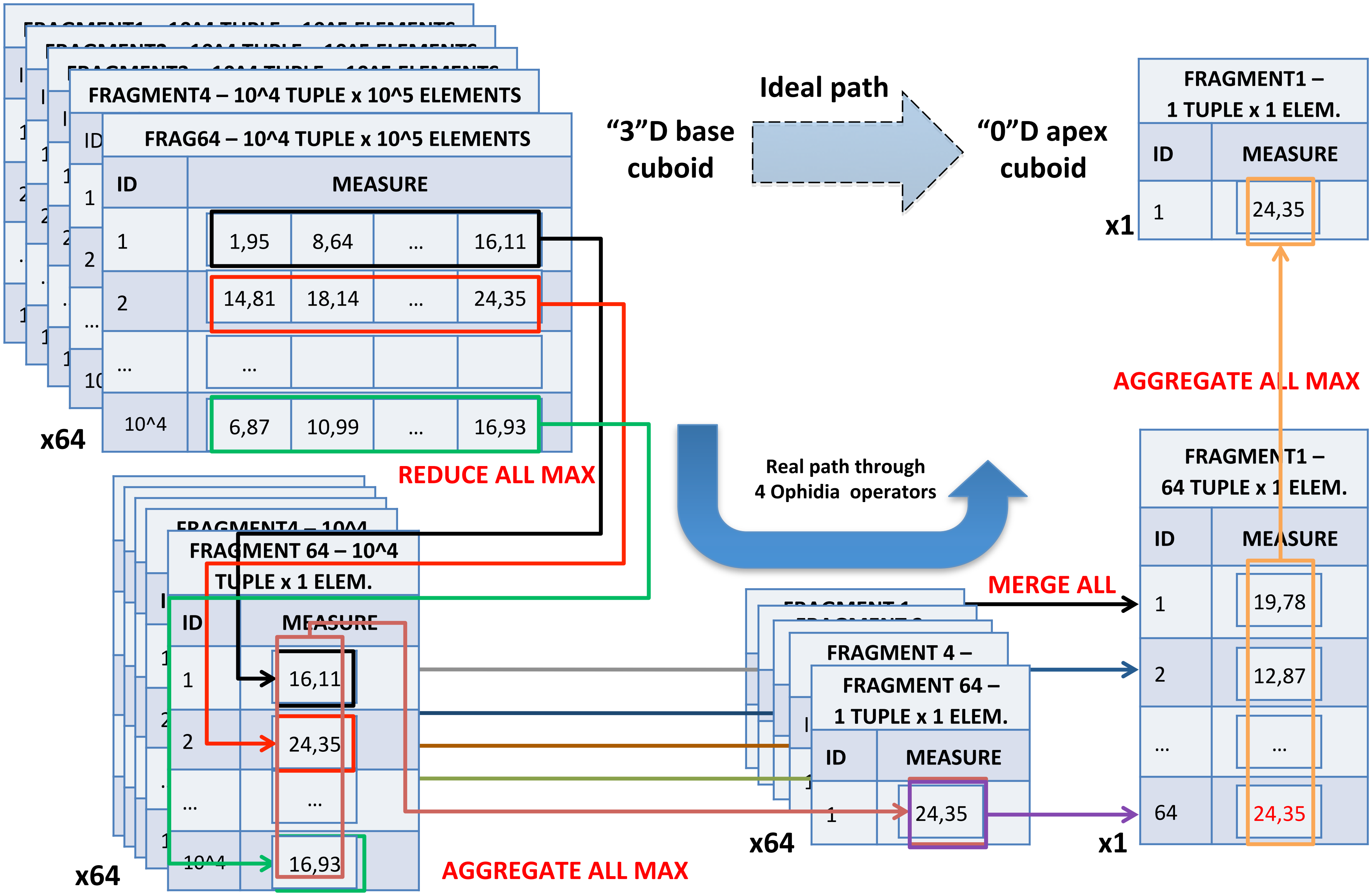
lat	lon	tos
39.500000	15.000000	1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20
39.500000	17.000000	287.3930664062, 286.8287048340, 286.5860595703, 286.9228210449, 288.5254516602, 292.3968200684, 295.8656921387, 297.2062072754, 295.7126464844
39.500000	19.000000	287.6926879883, 287.0508117676, 286.7896118164, 287.0781555176, 288.6802062988, 292.6882629395, 296.4769287109, 297.6632385254, 296.3418273926
40.500000	15.000000	1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20
40.500000	17.000000	287.1098632812, 286.5683593750, 286.2949829102, 286.5216674805, 288.0316772461, 291.7698974609, 295.4139709473, 296.8489685059, 295.4132995605
40.500000	19.000000	287.4010009766, 286.7818298340, 286.4914245605, 286.7260742188, 288.3006286621, 292.1842346191, 296.0237731934, 297.2694702148, 295.9751892090
41.500000	15.000000	1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20, 1.00000002e+20
41.500000	17.000000	286.5835876465, 286.0175781250, 285.7146911621, 285.9142761230, 287.4476623535, 291.1032104492, 294.7090454102, 296.0852355957, 294.7053222656
41.500000	19.000000	286.9717712402, 286.3946838379, 286.0617675781, 286.1446228027, 287.6101989746, 291.2955017090, 295.2700195312, 296.5146179199, 295.3194274902

```
Summary
-------
Selected 9 rows out of 9
```

The analytics framework: “metadata” operators



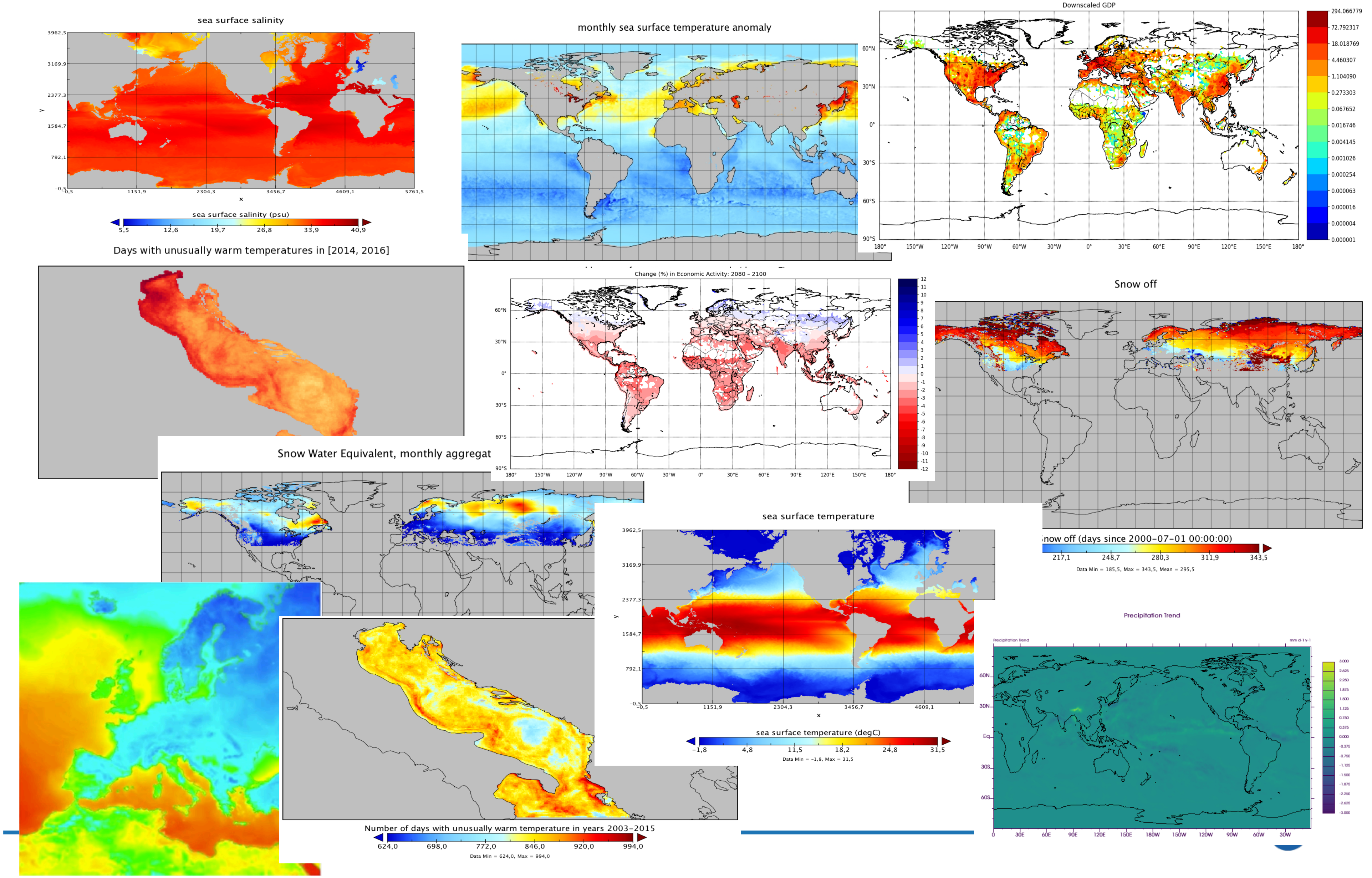
Pipelining analytics operators to reduce data



Ophidia architecture 2.0

Workflows management, python applications, in-memory analytics

Efficient support for advanced analytics experiments



Architecture evolution

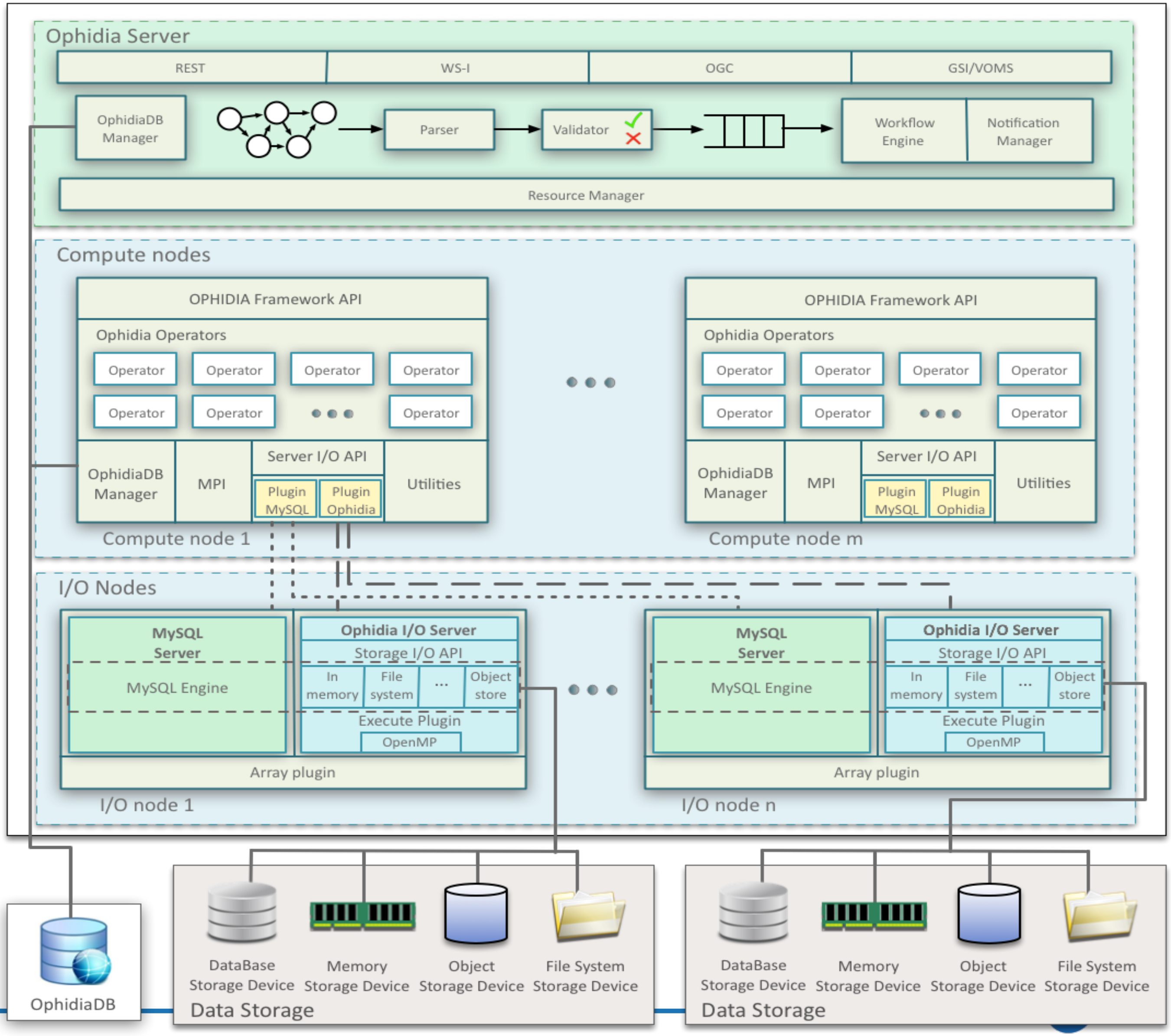
Workflow support on the server side

Separation of concerns between framework and I/O components

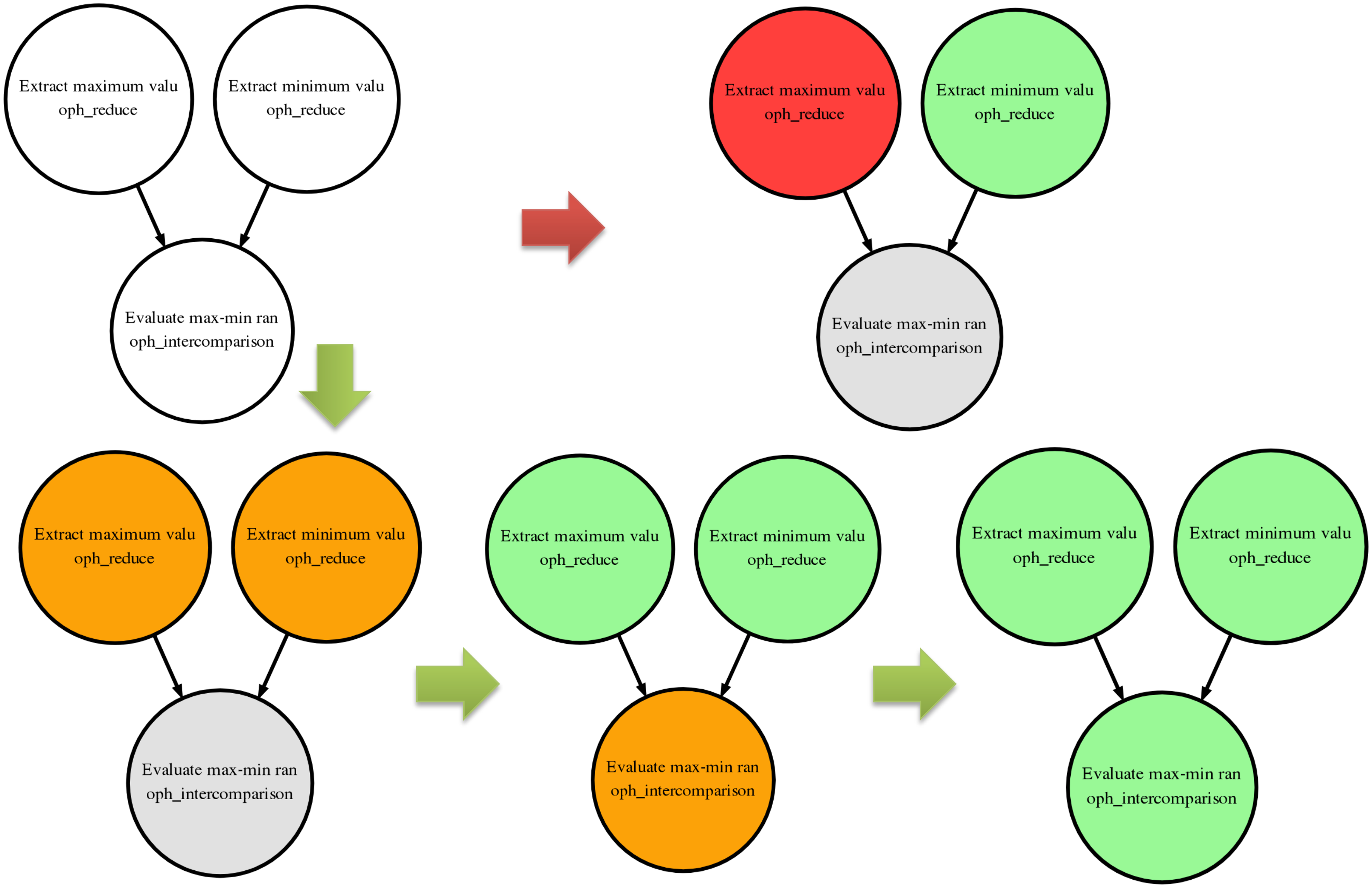
Support different **I/O servers**

Native I/O server with **parallel execution engine**

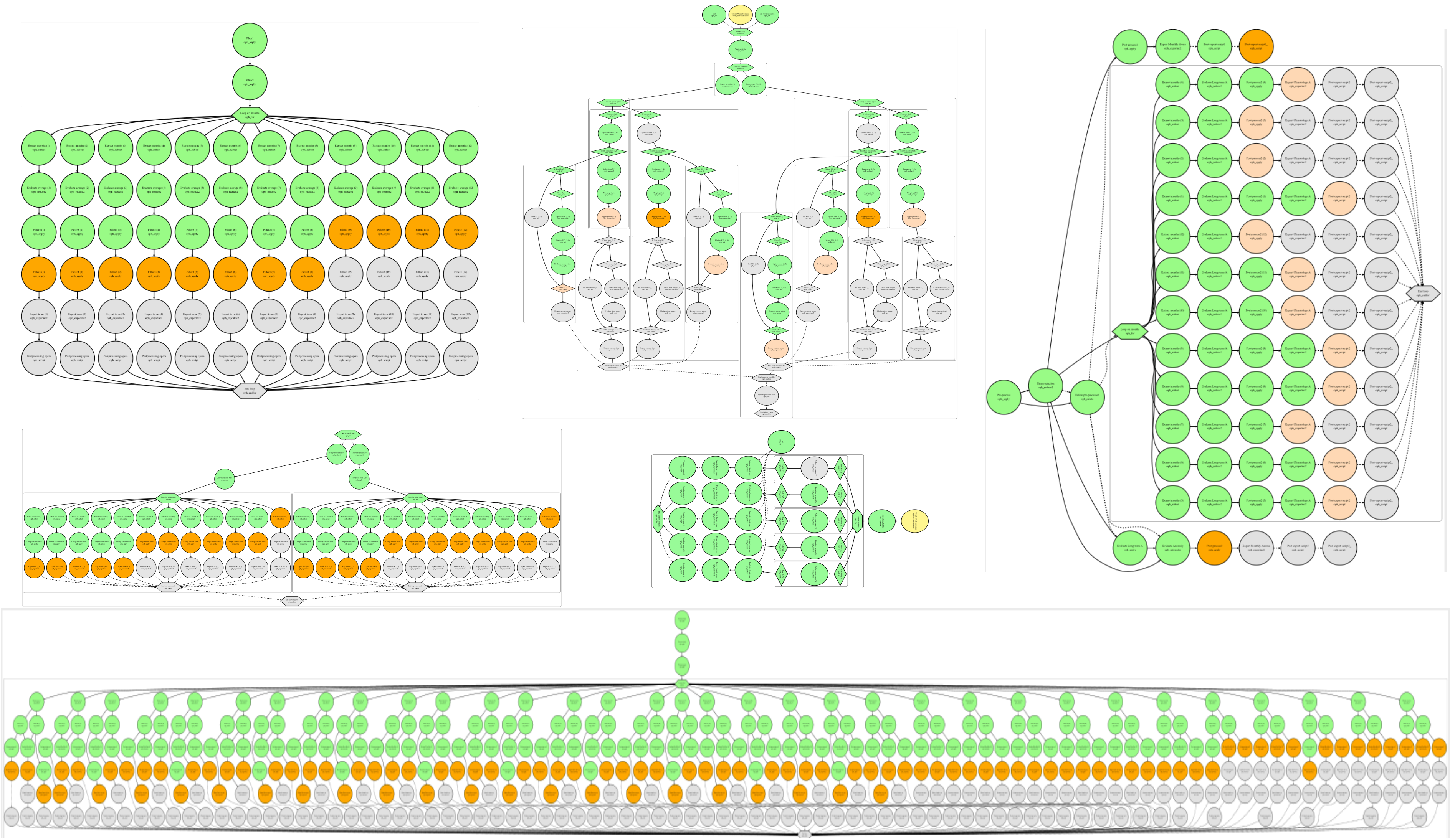
Multiple **storage systems** supported



Workflow support



Analytics workflows support and interfaces



Analytics workflows support and interfaces

Workflow Management

This group includes a number of flow control operators that could be used within an Ophidia workflow to implement complex data processing in batch mode. In particular, they implement several advanced features: setting of run-time variables, iterative and parallel interface, selection interface, interactive workflows, interleaving workflows, etc.

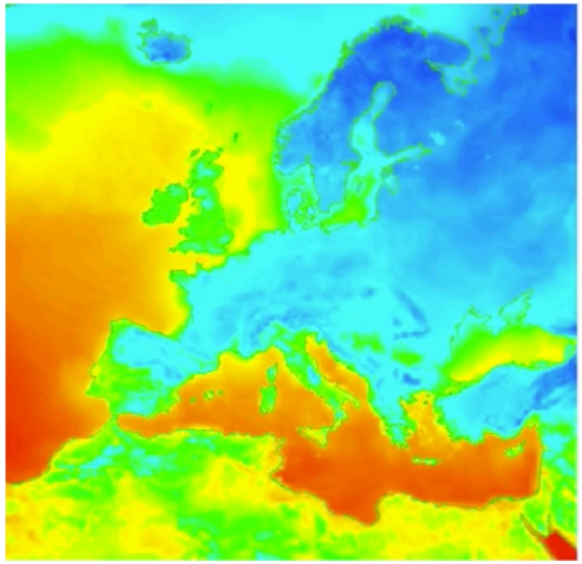
NAME	DESCRIPTION
OPH_ELSE	Start the last sub-block of a selection block "if".
OPH_ELSEIF	Start a new sub-block of a selection block "if".
OPH_ENDFOR	Close a loop "for".
OPH_ENDIF	Close a selection block "if".
OPH_FOR	Implement a loop "for".
OPH_IF	Open a "if" selection block.
OPH_INPUT	It sends commands or data to an interactive task.
OPH_SET	Set a parameter in the workflow environment.
OPH_WAIT	Wait until an event occurs.

Workflow I: climate indicators processing

- *In the CLIPC project, processing chains for data analysis are being implemented with Ophidia to compute* ***climate indicators***

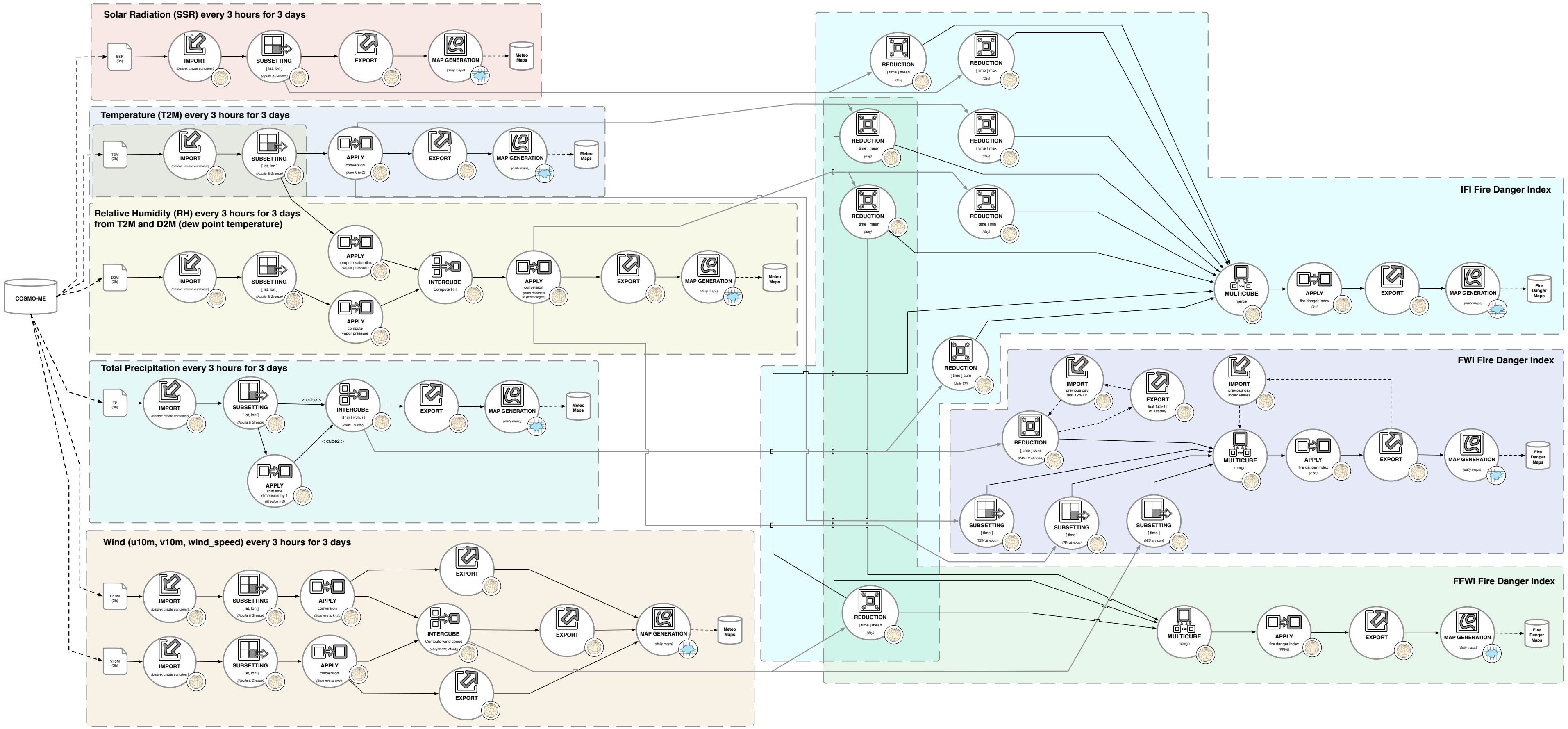
- ***First set of indicators*** *includes: TNn,* ***TNx****, TXn,* ***TXx***
  - *Input files: 12GBs (TasMin & TasMax)*
    - *TNx = max of the min temperatures*
    - *TXx = max of the max temperatures*

- ***Parallel approach***
  - *Inter-parallelism & Intra-parallelism*

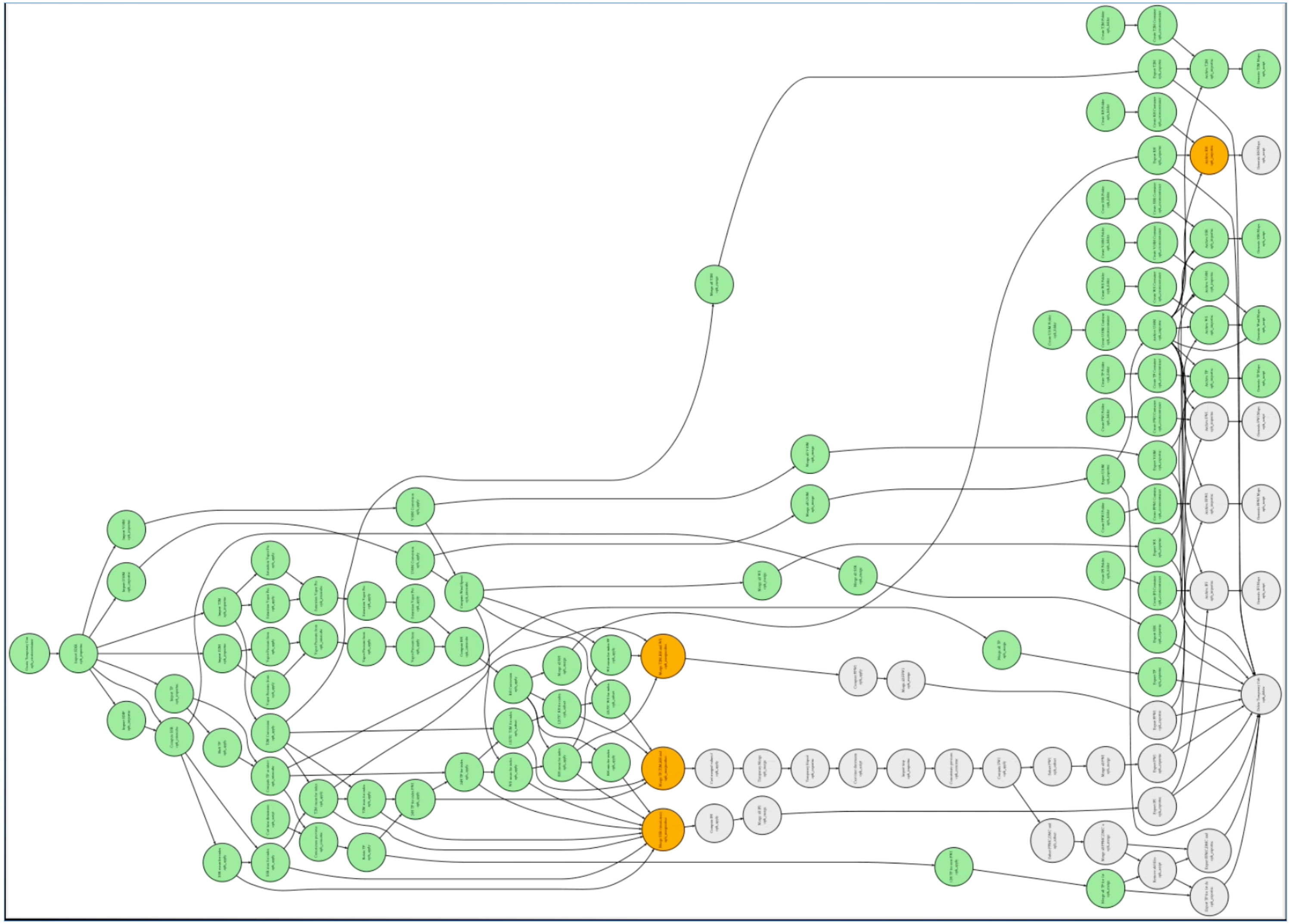


Workflow example II: fire danger analysis

OFIDIA main objective is to build a **cross-border operational fire danger prevention infrastructure** that advances the ability of regional stakeholders across Apulia and Ioannina Regions to **detect** and **fight forest wildfires**



Workflow example II: fire danger analysis
Runtime Execution



https://www.youtube.com/watch?v=vxbYF1Zhpuc&feature=youtu.be

Workflow example III: multi-model analytics

Cloud-enabled, distributed multi-model analytics experiment

2016 IEEE International Conference on Big Data (Big Data)

Distributed and cloud-based multi-model analytics experiments on large volumes of climate change data in the Earth System Grid Federation eco-system

S. Fiore[1], M. Płóciennik[2], C. Doutriaux[3], C. Palazzo[1], J. Boutte[3], T. Żok[2], D. Elia[1], M. Owsiak[2], A. D'Anca[1], Z. Shaheen[3], R. Bruno[4], M Fargetta[4], M. Caballer[5], G. Moltó[5], I. Blanquer[5], R. Barbera[4,6], M. David[7], G. Donvito[4], D. N. Williams[3], V. Anantharaj[8], D. Salomoni[4], and G. Aloisio[1,9]

[1]Euro-Mediterranean Center on Climate Change Foundation (CMCC), Italy
[2]Poznan Supercomputing and Networking Center (PSNC), Poland
[3]Lawrence Livermore National Laboratory (LLNL), California, USA
[4]Italian National Institute of Nuclear Physics (INFN), Italy
[5]Universitat Politècnica de València (UPV), Spain
[6] University of Catania, Italy
[7]Laboratório de Instrumentação e Fisica Experimental de Particulas (LIP), Portugal
[8]Oak Ridge National Laboratory (ORNL), Tennessee, USA
[9]University of Salento, Italy

Abstract—A case study on *climate models intercomparison data analysis* addressing several classes of multi-model experiments is being implemented in the context of the EU H2020 INDIGO-DataCloud project. Such experiments require the availability of large amount of data (multi-terabyte order) related to the output of several climate models simulations as well as the exploitation of scientific data management tools for large-scale data analytics. More specifically, the paper discusses in detail a use case on precipitation trend analysis in terms of requirements, architectural design solution, and infrastructural implementation. The experiment has been tested and validated on CMIP5 datasets, in the context of a large scale distributed testbed across EU and US involving three ESGF sites (LLNL, ORNL, and CMCC) and one central orchestrator site (PSNC).

Keywords-big analytics, workflow management, cloud computing, ESGF, INDIGO-DataCloud.

I. Introduction

The increased models resolution in the development of comprehensive Earth System Models is rapidly leading to very large climate simulations output that pose significant scientific data management challenges in terms of data sharing, processing, analysis, visualization, preservation, curation, and archiving [1-3].

In this domain, large scale global experiments for climate model interconparison (CMIP) have led to the development of the Earth System Grid Federation (ESGF [4-5]), a federated data infrastructure involving a large set of data providers/modelling centers around the globe, which includes the European contribution - regarding the ENES [6] community – through the IS-ENES project.

From an infrastructural standpoint, ESGF provides a production-level support for search & discovery, browsing and access to climate simulation data and observational data products. ESGF has been serving the Coupled Model Intercomparison Project Phase 5 (CMIP5) experiment, providing access to 2.5PB of data for the Intergovernmental Panel on Climate Change (IPCC) [7] Assessment Reports 5 [8], based on consistent metadata catalogues. More precisely, the Coupled Model Intercomparison Project (CMIP) has been established by the Working Group on Coupled Modelling [9] (WGCM) under the World Climate Research Programme [10] (WCRP).

It provides a community-based infrastructure in support of climate model diagnosis, validation, interconparison, documentation and data access. This framework enables a diverse community of scientists to analyse General Circulation Models (GCMs) in a systematic fashion, a process that serves to facilitate models improvement.

CMIP5 has promoted a standard set of model simulations in order to:

- evaluate how realistic the models are in simulating the recent past;
- provide projections of future climate change on two time scales, near term (out to about 2035) and long term (out to 2100 and beyond); and
- understand some of the factors responsible for differences in model projections, including quantifying some key feedbacks such as those involving clouds and the carbon cycle.

In such a context, running a multi-model data analysis experiment is very challenging, as it requires the availability of large amount of data (multi-terabyte order) related to multiple climate models simulations as well as scientific data management tools for large-scale data analytics.

The remainder of this work is organized as it follows. Section II provides the current workflow for the multi-model climate data analysis in the CMIP context, whereas Section III presents the paradigm shift needed to address such large-

978-1-4673-9005-7/16/$31.00 ©2016 IEEE 2911

Big Data Challenges, Research, and Technologies in the Earth and Planetary Sciences

A workshop to be held Monday December 5th at the 2016 IEEE International Big Data Conference

INDIGO - DataCloud

- *A first experiment across sites was demonstrated at the 1st INDIGO Review, November 2016 in Bologna*
- *Strong synergy with the ESGF CWT Roadmap*
- *International collaboration across the Atlantic*

S. Fiore, M. Plóciennik, et al.: Distributed and cloud-based multi-model analytics experiments on large volumes of climate change data in the Earth System Grid Federation eco-system. BigData 2016: 2911-2918

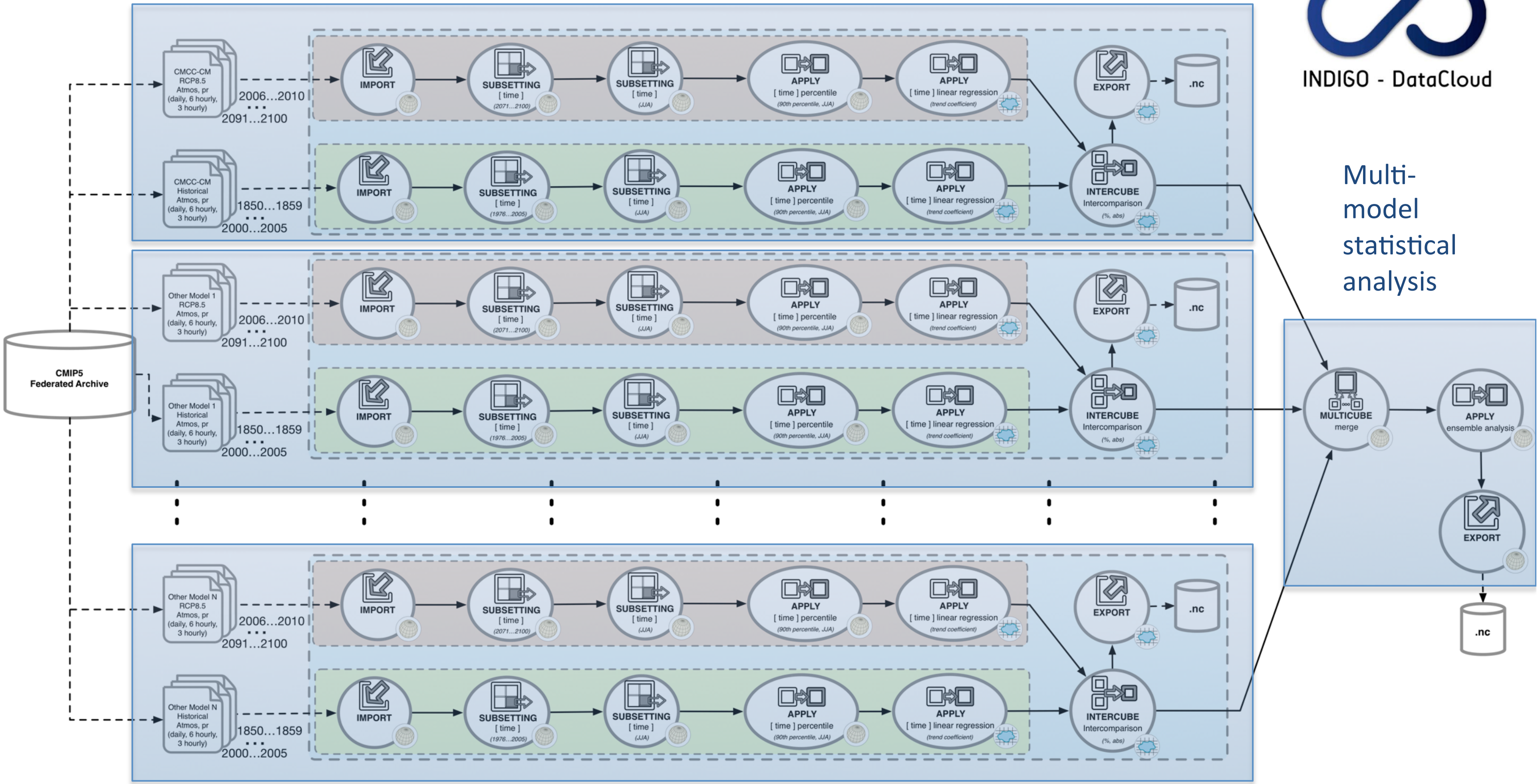
INDIGO-DataCloud

- **An H2020 project** approved in January 2015 in the EINFRA-1-2014 call
  - 11.1M€, 30 months (**from April 2015 to September 2017**)
- **Who**: **26 European partners** in 11 European countries
  - Coordination by the Italian National Institute for Nuclear Physics (INFN)
  - Including developers of distributed software, industrial partners, research institutes, universities, e-infrastructures
- **What**: **develop an open source Cloud platform** for computing and data ("DataCloud") tailored to science.
- **For**: **multi-disciplinary scientific communities**
  - E.g. structural biology, earth science, physics, bioinformatics, cultural heritage, astrophysics, life science, climatology
- **Where**: deployable on **hybrid (public or private) Cloud infrastructures**
  - INDIGO = **IN**tegrating **D**istributed data **I**nfrastructures for **G**lobal Expl**O**itation
- **Why**: answer to the technological **needs of scientists** seeking to easily exploit distributed Cloud/Grid compute and data resources.

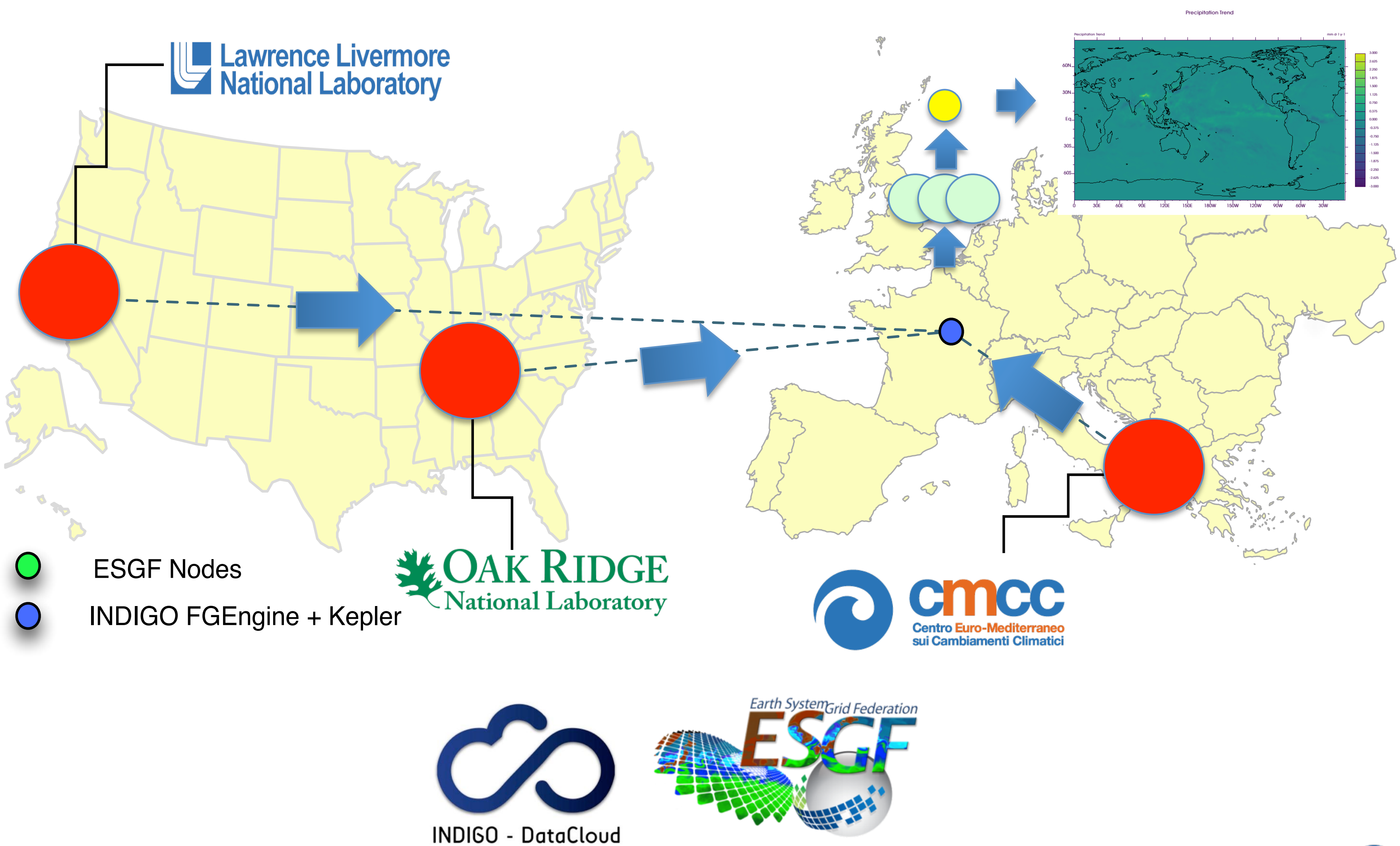
High-level view of the multi-model „precipitation trend analysis” experiment

Single model precipitation trend analysis

Multi-model statistical analysis

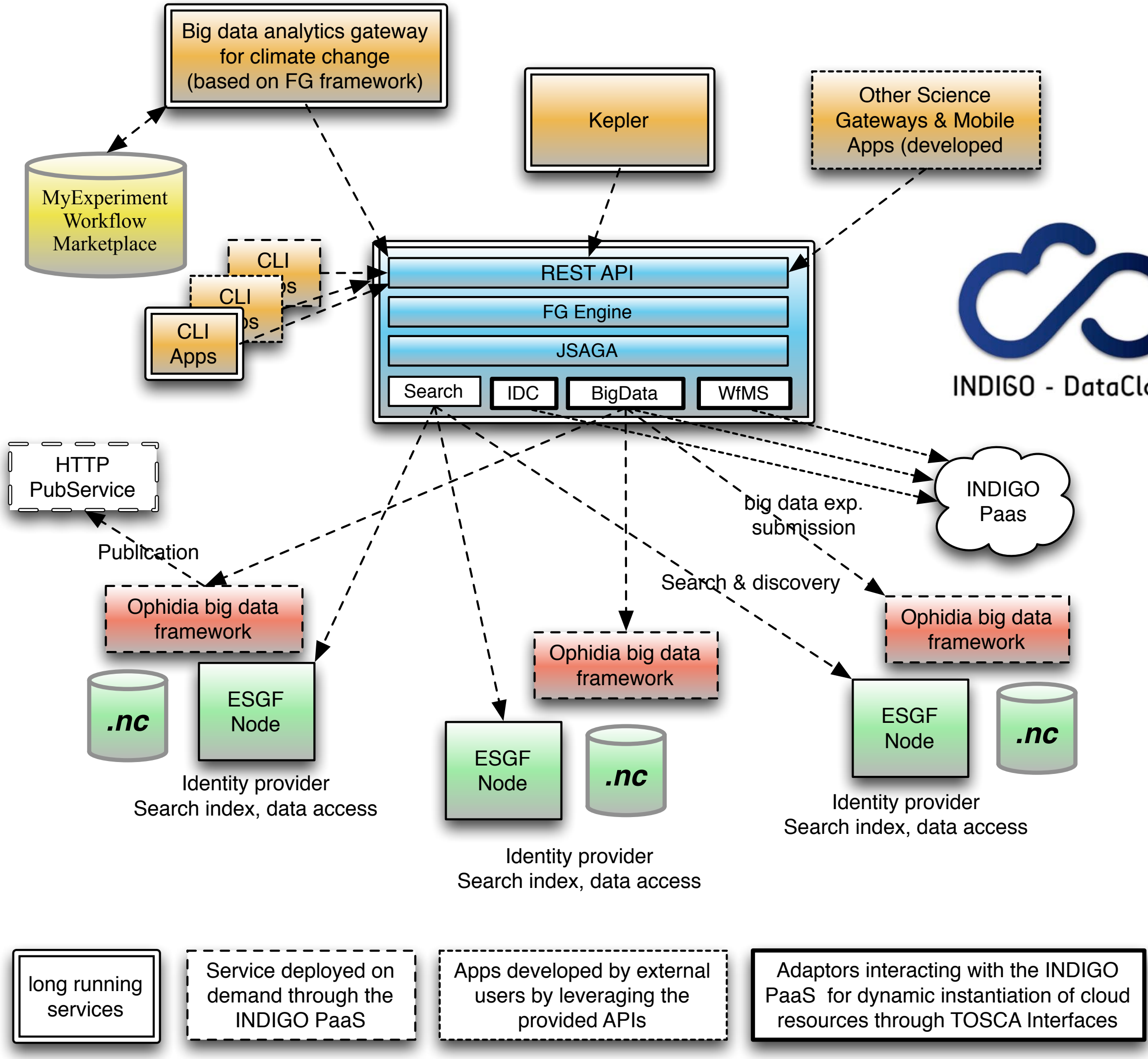


CMIP5 scientific data analysis workflow in ESGF



INDIGO-DataCloud architectural solution

- Distributed experiments for climate data analysis
- Server-side, parallel processing
- Two-level workflow strategy to orchestrate large scale experiments
- Interoperability with ESGF
- Access through different clients
  - Kepler
  - Science Gateway
- Interactive and batch scenarios



INDIGO - DataCloud

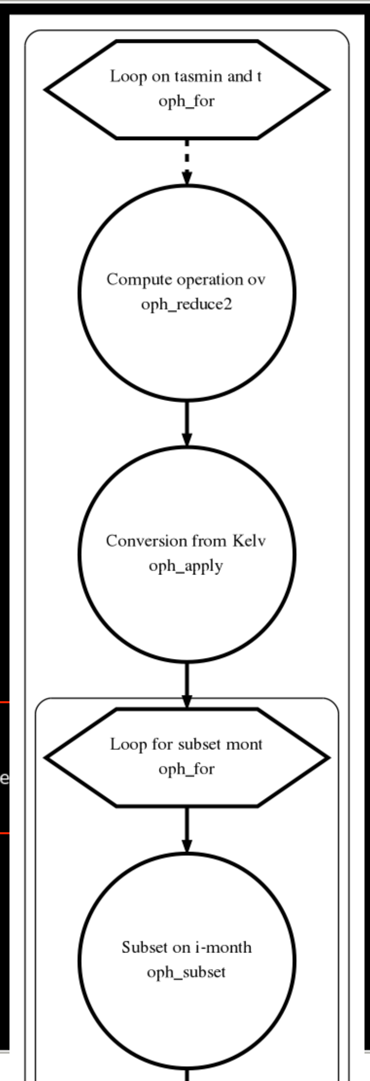
The paradigm shift proposed & exploited in INDIGO-DataCloud

Behind the scene: workflow JSON representation

ophrpm@ophidiarpm:~/workflow

```
"tasks": [
    {
        "name": "Loop on tasmin and tasmax cubes",
        "operator": "oph_for",
        "arguments": [ "name=cube","counter=1:2", "values=${1}|${2}", "parallel=yes" ]
    },
            {
                    "name": "Compute operation over time",
                    "operator": "oph_reduce2",
                    "arguments": [
                            "cube=@{cube}",
                            "dim=time",
                            "concept_level=M",
                            "midnight=00",
                            "operation=$3",
                            "container=tmp"
                    ],
                    "dependencies": [
                            { "task": "Loop on tasmin and tasmax cubes" }
                    ]
            },
            {
                    "name": "Conversion from Kelvin to Celsius degrees",
                    "operator": "oph_apply",
                    "arguments": [
                            "query=oph_sum_scalar('oph_float','oph_float',measure,-273.15)"
                    ],
                    "dependencies": [{
                            "task": "Compute operation over time",
                            "type": "single"
                    }]
            },
            {
                    "name": "Loop for subset months",
                    "operator": "oph_for",
                    "arguments": [ "name=index","counter=1:12", "values=Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|N ... lle
                    "dependencies": [ { "task":"Conversion from Kelvin to Celsius degrees", "type": "single" } ]
            },
                    {
                            "name": "Subset on i-month",
                            "operator": "oph_subset",
                            "arguments": [
                                    "subset_dims=time",
                                    "subset_filter=&index:12:end"
                            ],
                            "dependencies": [
--More--(65%)
```

ophrpm@ophidiarpm:~/workflow



***Youtube video**: https://www.youtube.com/watch?v=PTZkw60YCNU*

Workflow submission

ophrpm@ophidiarpm:~/devel/oph-client/res

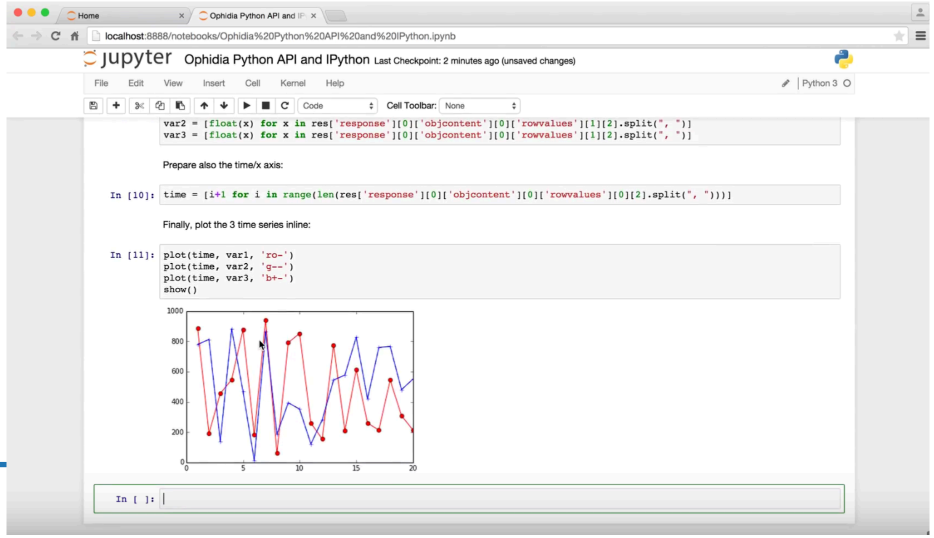
ophrpm@ophidiarpm:~/workflow

```
[37..6380] >>
[37..6380] >> ./Tind_loop.json http://193.204.199.174/ophidia/29/2046 http://193.204.199.174/ophidia/30/2047 max
[JobID]:
http://193.204.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3144

[37..6380] >> view 247
[247] ./Tind_loop.json http://193.204.199.174/ophidia/29/2046 http://193.204.199.174/ophidia/30/2047 max [http://193.204.199.174/ophidia/sessions/376699238311302
232511449455166146380/experiment?247#3144]

[Response]:
Workflow Status
---------------

OPH_STATUS_COMPLETED

Workflow Progress
-----------------

+===========================+=======================+
| NUMBER OF COMPLETED TASKS | TOTAL NUMBER OF TASKS |
+===========================+=======================+
| 82                        | 82                    |
+===========================+=======================+

Workflow Task List
------------------

+========================================================+======================+========+=======+===========+============================+=====+=============+
| OPH JOB ID                                             | SESSION CODE         | WORKFL | MARKE | PARENT MA | TASK NAME                  | TYP | EXIT STATUS |
|                                                        |                      | OW ID  | R ID  | RKER ID   |                            | E   |             |
+========================================================+======================+========+=======+===========+============================+=====+=============+
| http://193.204.199.174/ophidia/sessions/37669923831130 | 37669923831130223251 | 247    | 3145  | 3144      | Loop on tasmin and tasmax  | SIM | OPH_STATUS_ |
| 2232511449455166146380/experiment?247#3145             | 1449455166146380     |        |       |           |  cubes                     | PLE | COMPLETED   |
|--------------------------------------------------------|----------------------|--------|-------|-----------|----------------------------|-----|-------------|
| http://193.204.199.174/ophidia/sessions/37669923831130 | 37669923831130223251 | 247    | 3146  | 3144      | Compute operation over ti  | SIM | OPH_STATUS_ |
| 2232511449455166146380/experiment?247#3146             | 1449455166146380     |        |       |           | me (1)                     | PLE | COMPLETED   |
|--------------------------------------------------------|----------------------|--------|-------|-----------|----------------------------|-----|-------------|
| http://193.204.199.174/ophidia/sessions/37669923831130 | 37669923831130223251 | 247    | 3147  | 3144      | Compute operation over ti  | SIM | OPH_STATUS_ |
| 2232511449455166146380/experiment?247#3147             | 1449455166146380     |        |       |           | me (2)                     | PLE | COMPLETED   |
|--------------------------------------------------------|----------------------|--------|-------|-----------|----------------------------|-----|-------------|
| http://193.204.199.174/ophidia/sessions/37669923831130 | 37669923831130223251 | 247    | 3148  | 3144      | Conversion from Kelvin to  | SIM | OPH_STATUS_ |
| 2232511449455166146380/experiment?247#3148             | 1449455166146380     |        |       |           |  Celsius degrees (1)       | PLE | COMPLETED   |
|--------------------------------------------------------|----------------------|--------|-------|-----------|----------------------------|-----|-------------|
| http://193.204.199.174/ophidia/sessions/37669923831130 | 37669923831130223251 | 247    | 3149  | 3144      | Conversion from Kelvin to  | SIM | OPH_STATUS_ |
| 2232511449455166146380/experiment?247#3149             | 1449455166146380     |        |       |           |  Celsius degrees (2)       | PLE | COMPLETED   |
|--------------------------------------------------------|----------------------|--------|-------|-----------|----------------------------|-----|-------------|
```

Programmatic access through the PyOphidia class

- ✓ **PyOphidia** provides a Python interface to submit commands to the Ophidia Server and to retrieve/deserialize the results
- ✓ Two classes implemented:
  - ✓ **Client class**: connect to the server, navigate into the ophidia file system, submit workflows, manage sessions, etc.
  - ✓ **Cube class**: manipulate cubes (reduce, subset, operations between cubes, intercomparison, etc.), get information on cubes (schema, dimensions, metadata, etc.)

```
class Cube():
    """Cube(container='-', cwd=None, exp_dim='auto', host_partition='auto', imp_dim='auto', measure=None, src_path=None, cdd=None, compressed='no',
            exp_concept_level='c', filesystem='auto', grid='-', imp_concept_level='c', import_metadata='no', check_compliance='no', offset=0,
            ioserver='mysql_table', ncores=1, ndb=1, ndbms=1, nfrag=0, nhost=0, subset_dims='none', subset_filter='all', time_filter='yes'
            subset_type='index', exec_mode='sync', base_time='1900-01-01 00:00:00', calendar='standard', hierarchy='oph_base', leap_month=2,
            leap_year=0, month_lengths='31,28,31,30,31,30,31,31,30,31,30,31', run='yes', units='d', vocabulary='-', description='-', schedule=0,
            pid=None, check_grid='no', display=False) -> obj
       or Cube(pid=None) -> obj

    Attributes:
        pid: cube PID
        creation_date: creation date of the cube
        measure: name of the variable imported into the cube
        measure_type: measure data type
        level: number of operations between the original imported cube and the actual cube
        nfragments: total number of fragments
        source_file: parent of the actual cube
        hostxcube: number of hosts associated with the cube
        dbmsxhost: number of DBMS instances on each host
        dbxdbms: number of databases for each DBMS
        fragxdb: number of fragments for each database
        rowsxfrag: number of rows for each fragment
        elementsxrow: number of elements for each row
        compressed: 'yes' for a compressed cube, 'no' otherwise
        size: size of the cube
        nelements: total number of elements
        dim_info: list of dict with information on each cube dimension

    Class Attributes:
        client: instance of class Client through which it is possible to submit all requests
```



PyOphidia release

Search projects

Help Donate Log in Register

PyOphidia 1.6.0

```
pip install PyOphidia
```

Latest version

Last released: About 6 days ago

Python bindings for the Ophidia Data Analytics Platform

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

View statistics for this project via Libraries.io, or by using Google BigQuery

Project description

PyOphidia is a GPLv3-licensed Python package for interacting with the Ophidia framework.

It is an alternative to Oph_Term, the Ophidia no-GUI interpreter component, and a convenient way to submit SOAP HTTPS requests to an Ophidia server or to develop your own application using Python.

It runs on Python 2.7, 3.3, 3.4 and 3.5, has no Python dependencies and is pure-Python code. It requires a running Ophidia instance for client-server interactions. The latest PyOphidia version (v1.6.0) is compatible with Ophidia v1.3.0.

It provides 2 main modules:

- client.py: generic *low level* class to submit any type of requests (simple tasks and workflows), using SSL and SOAP with the client ophsubmit.py;
- cube.py: *high level* cube-oriented class to interact directly with cubes, with several methods wrapping the operators.

Installation

https://pypi.org/project/PyOphidia/

PyOphidia applications: Jupyter notebooks

Import PyOphidia and connect to server instance

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

Import data and extract a single time series

In [ ]:
```
mycube = cube.Cube.importnc(src_path='/public/data/tos_O1_2001-2002.nc',measure='tos',imp_dim='time',ncores=5)
mycube2 = mycube.subset2(subset_dims="lat|lon",subset_filter="0:1|0:1",ncores=5)
data = mycube2.export_array()
```

Plot time series

In [ ]:
```
import matplotlib.pyplot as plt
y = data['measure'][0]['values'][0][:]
x = data['dimension'][2]['values'][:]
plt.figure(figsize=(11, 3), dpi=100)
plt.plot(x, y)

plt.ylabel(data['measure'][0]['name'] + " (degK)")
plt.xlabel("Days since 2001/01/01")
plt.title('Sea Surface Temperature (point 0.5, 1)')
plt.show()
```

Convert from Kelvin to Celsius degrees

In [ ]:
```
mycube3 = mycube2.apply(query="oph_sum_scalar('OPH_FLOAT','OPH_FLOAT',measure,-273.15)",description="celsius")
data = mycube3.export_array()
```

Plot time series

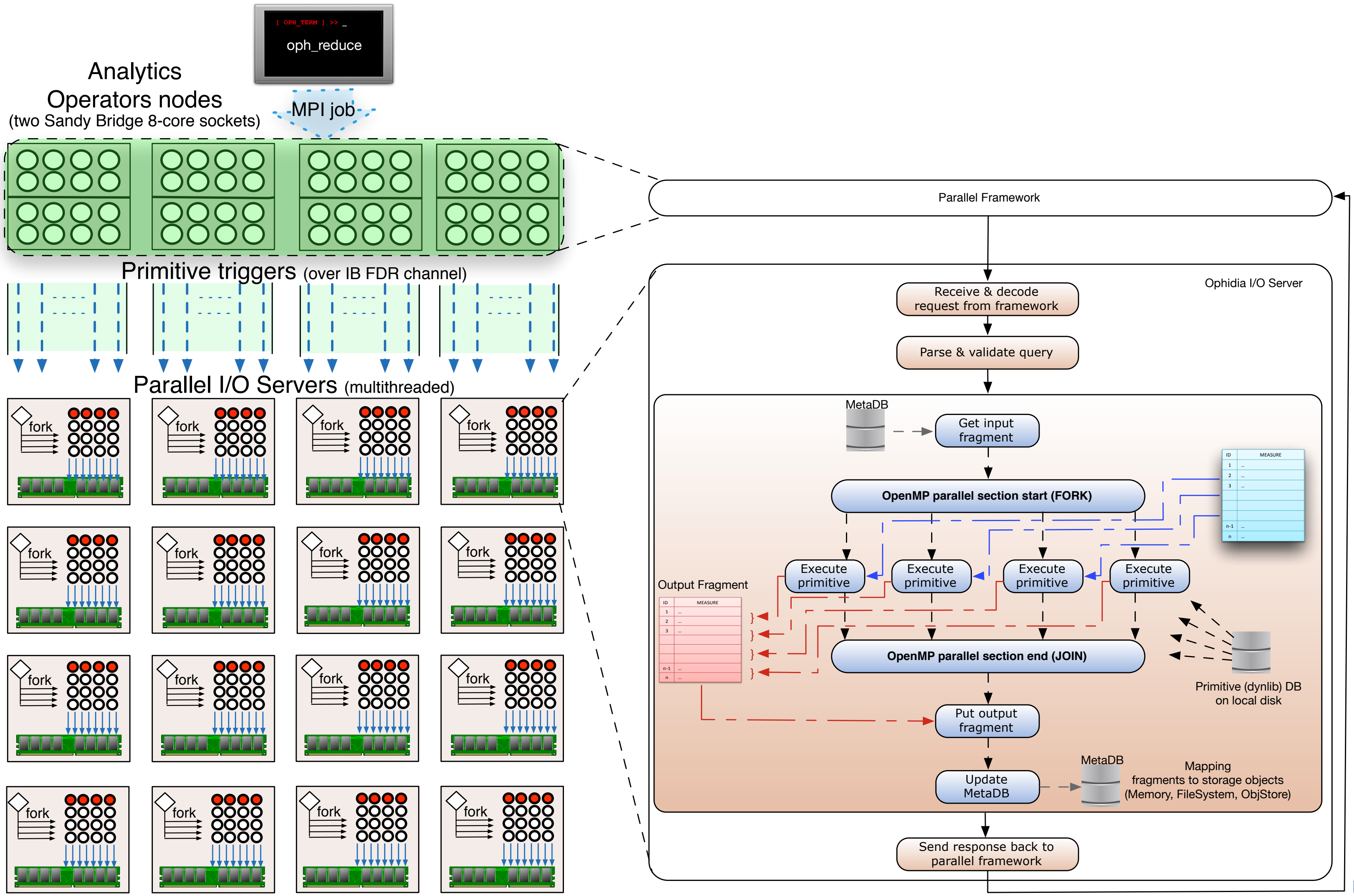
In [ ]:
```
y = data['measure'][0]['values'][0][:]
x = data['dimension'][2]['values'][:]
plt.figure(figsize=(11, 3), dpi=100)
plt.plot(x, y)

plt.ylabel(data['measure'][0]['name'] + " (degC)")
plt.xlabel("Days since 2001/01/01")
plt.title('Sea Surface Temperature (point 0.5, 1)')
plt.show()
```

Native Ophidia I/O server

The I/O server provides a native solution for the scientific domain applications. The requirements for the Ophidia I/O server are:

- run **data analytics tasks in-memory** taking advantage of the lower latency
- **binary array-oriented engine** to efficiently process scientific multidimensional data
- interact directly with the storage layer to **exploit data locality**
- exploit **parallelism at the array-level**
- **NoSQL approach** based on key-value store providing a **declarative query language** (SQL-like)
- guarantee extensibility and interoperability of the I/O server to **support multiple storage back-ends**

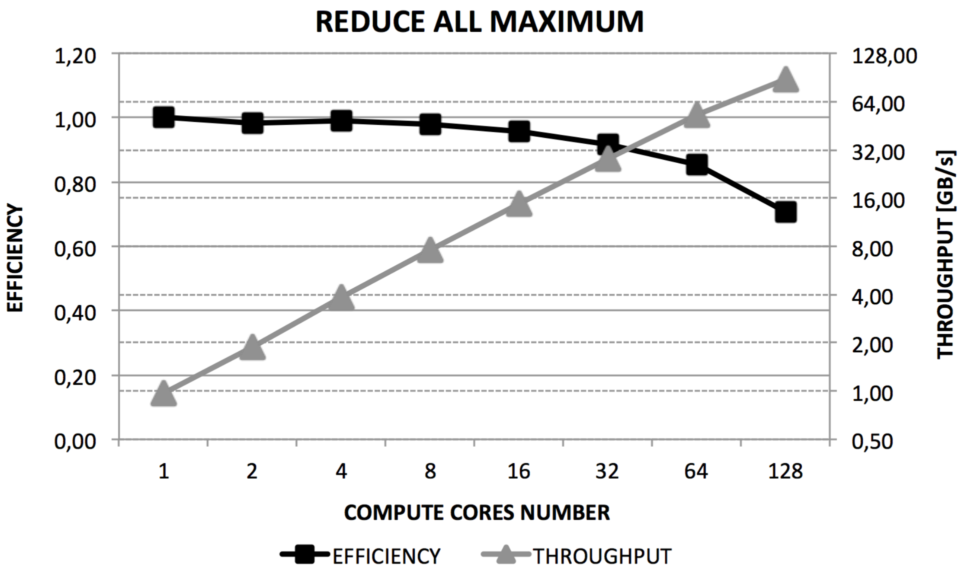
Parallel support: in-depth view of the parallel reduce



Experimental results (in-memory I/O server)

Execution time is measured by scaling up the number of parallel tasks
Two metrics are evaluated:

- efficiency (speedup/computational resources)
- throughput (data processed/time unit)

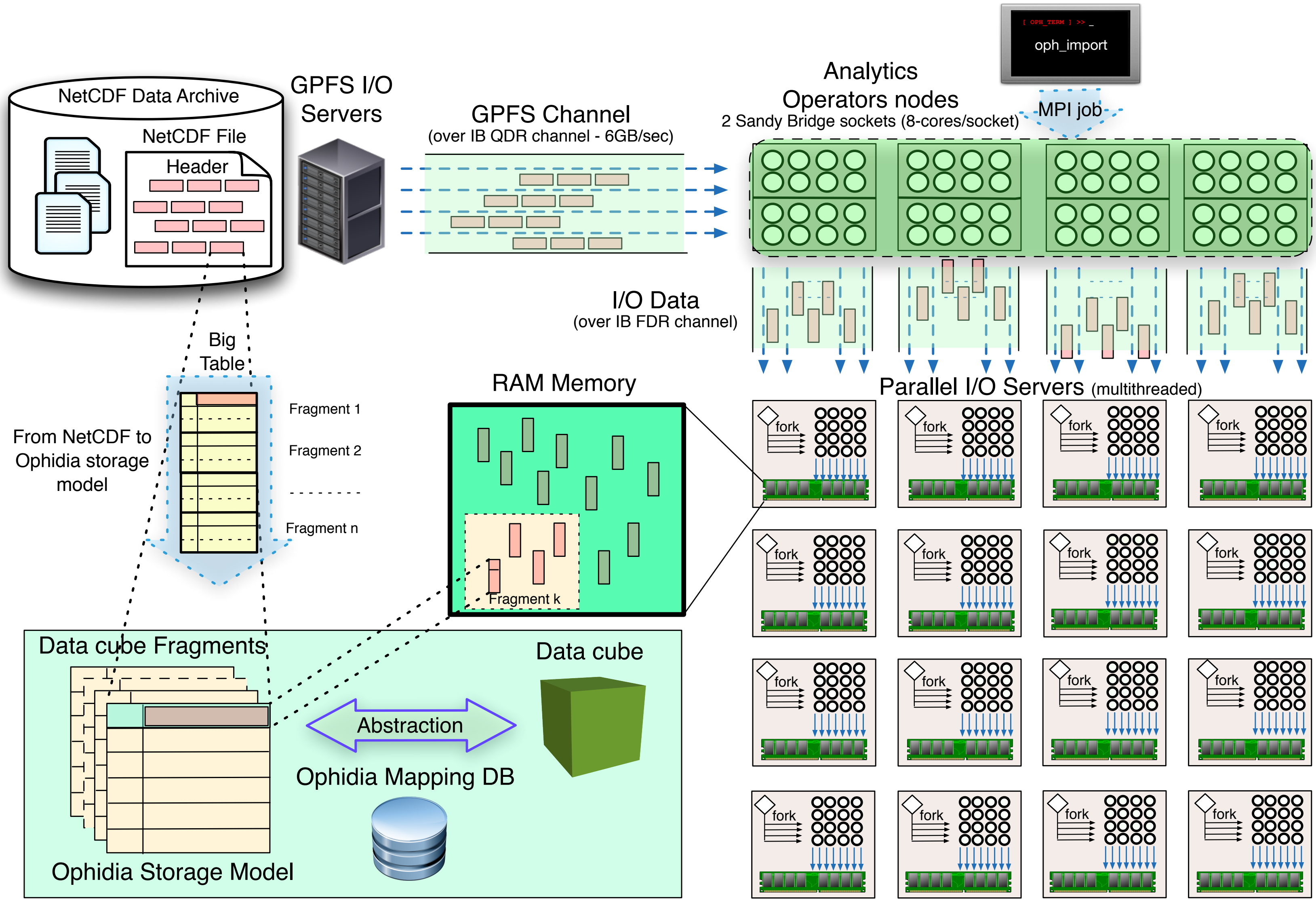


CORES NUMBER	EXECUTION TIME [s]	EFFICIENCY	THROUGHPUT [GB/s]
1	388,50	1,00	0,97
2	197,51	0,98	1,90
4	97,96	0,99	3,83
8	49,52	0,98	7,57
16	25,39	0,96	14,77
32	13,22	0,92	28,36
64	7,11	0,85	52,72
128	**4,29**	0,71	87,47

3D dataset, 375GB, 2.1M time series, 24K elements each (50 Billions elements)
8 nodes, 16 cores each, 128 cores in total
Max computation over time dimension, 2D result (map)

With 128 cores it is around 30x faster than MySQL I/O engine!
Full benchmark is ongoing on the Athena Cluster at CMCC SCC

Parallel import and the new import2 (10X speedup)

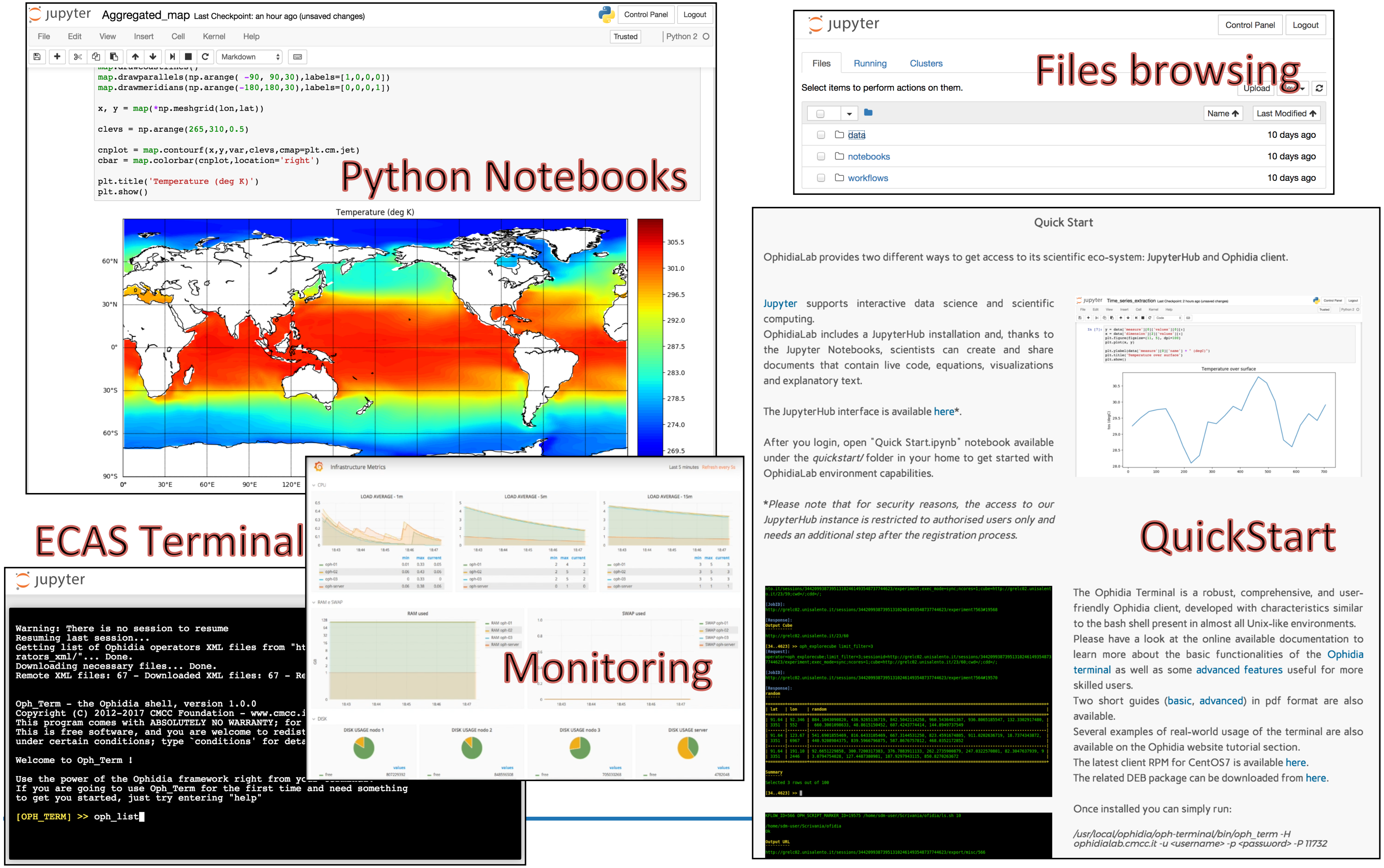


ECASLab in the EOSC-hub context

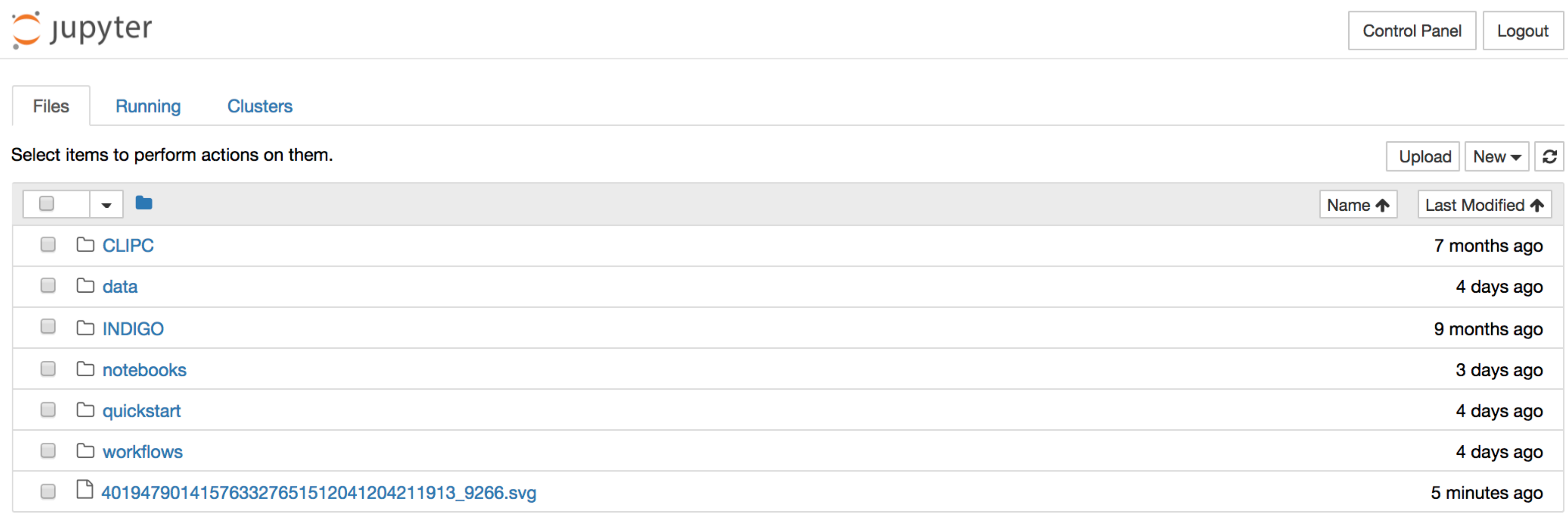
ECASLab: a user-oriented environment for data analysis and visualization

- ✔ ***ECASLab*** *is an integrated scientific environment for scientific data management*
- ✔ *It provides a ready-to-use multi-node* ***ECAS (ENES Climate Analytics Service)*** *to perform data analytics on scientific datasets*
- ✔ *Currently setup at at CMCC (Italy) and DKRZ (Germany)*
- ✔ *It integrates data,* ***analysis and visualization tools*** *in a user-friendly environment accessible with light-weight clients (i.e. a desktop bash-like client and a web GUI)*
- ✔ *It exposes a* ***JupyterHub*** *service to create, execute and share Jupyter notebooks (Python-based) supporting live-code and visualization*
- ✔ *File system navigation, file editing, upload and download supported via web*
- ✔ *Released on May 2017, with an initial set of services:*
  - ✔ *Simple quick start & registration form available*
  - ✔ *JupyterHub, OPeNDAP/THREDDS/IDV, ECAS Terminal*
  - ✔ *Monitoring system based on Grafana*
  - ✔ *Besides PyOphidia Several Python libraries available for analysis & visualization*
  - ✔ *Workflow IDE (alpha release)*

ECASLab in a nutshell



ECASLab: Jupyter user local folder



ECASLab: Jupyter notebooks

jupyter Time_series_extraction (read only)

Control Panel Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 2

Markdown

Import PyOphidia and connect to server instance

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

Import data and extract a single time series

In [ ]:
```
mycube = cube.Cube.importnc(src_path='/public/data/tos_O1_2001-2002.nc',measure='tos',imp_dim='time',ncores=5)
mycube2 = mycube.subset2(subset_dims="lat|lon",subset_filter="0:1|0:1",ncores=5)
data = mycube2.export_array()
```

Plot time series

In [ ]:
```
import matplotlib.pyplot as plt
y = data['measure'][0]['values'][0][:]
x = data['dimension'][2]['values'][:]
plt.figure(figsize=(11, 3), dpi=100)
plt.plot(x, y)

plt.ylabel(data['measure'][0]['name'] + " (degK)")
plt.xlabel("Days since 2001/01/01")
plt.title('Sea Surface Temperature (point 0.5, 1)')
plt.show()
```

Convert from Kelvin to Celsius degrees

In [ ]:
```
mycube3 = mycube2.apply(query="oph_sum_scalar('OPH_FLOAT','OPH_FLOAT',measure,-273.15)",description="celsius")
data = mycube3.export_array()
```

Plot time series

In [ ]:
```
y = data['measure'][0]['values'][0][:]
x = data['dimension'][2]['values'][:]
plt.figure(figsize=(11, 3), dpi=100)
plt.plot(x, y)

plt.ylabel(data['measure'][0]['name'] + " (degC)")
```

ECASLab: ECAS Terminal (from Jupyter)

jupyter

Control Panel | Logout

```
| c | tos_01_2001-2002.nc | https://ophidialab.cmcc.it/ophidia/2017/260995 | celsius |

Execution time: 0.11 seconds
[40..1913] >> oph_
oph_aggregate            oph_duplicate     oph_if            oph_manage_session     oph_rollup
oph_aggregate2           oph_else          oph_importfits    oph_merge              oph_script
oph_apply                oph_elseif        oph_importnc      oph_mergecubes         oph_search
oph_cancel               oph_endfor        oph_importnc2     oph_mergecubes2        oph_service
oph_cluster              oph_endif         oph_importnc4     oph_metadata           oph_set
oph_concatnc             oph_explorecube   oph_importnc5     oph_movecontainer      oph_showgrid
oph_containerschema      oph_explorenc     oph_importnc6     oph_operators_list     oph_split
oph_createcontainer      oph_exportfits    oph_importsac     oph_permute            oph_subset
oph_cubeelements         oph_exportnc      oph_input         oph_primitives_list    oph_subset2
oph_cubeio               oph_exportnc2     oph_instances     oph_publish            oph_tasks
oph_cubeschema           oph_folder        oph_intercube     oph_randcube           oph_unpublish
oph_cubesize             oph_for           oph_list          oph_reduce             oph_wait
oph_delete               oph_fs            oph_log_info      oph_reduce2
oph_deletecontainer      oph_get_config    oph_loggingbk     oph_restorecontainer
oph_drilldown            oph_hierarchy     oph_man           oph_resume
[40..1913] >> oph_cube
oph_cubeelements  oph_cubeio        oph_cubeschema    oph_cubesize
[40..1913] >> oph_cubeschema
[Request]:
operator=oph_cubeschema;sessionid=https://ophidialab.cmcc.it/ophidia/sessions/401947901415763327651512041204211913/experiment;exec_mode=sync;cube=https://ophidialab.cmcc.it/ophidia/2017/260995;cwd=/;cdd=/home/sfiore;host_partition=test;

[JobID]:
https://ophidialab.cmcc.it/ophidia/sessions/401947901415763327651512041204211913/experiment?241#9269

[Response]:
Datacube Information
```

PID	CREATION DATE	MEASURE	MEASURE TYPE	LEVEL	NUMBER OF FRAGMENTS	SOURCE FILE
https://ophidialab.cmcc.it/ophidia/2017/260995	2018-06-25 03:15:30	tos	FLOAT	2	1	

Datacube Additional Information

DESCRIPTION	HOST x CUBE	DBMS x HOST	DATABASES x DBMS	FRAGMENTS x DATABASE	ROWS x FRAGMENT	ELEMENTS x ROW	COMPRESSED	CUBE SIZE	UNIT	NUMBER OF ELEMENTS
celsius	1	1	1	1	1	24	no			24

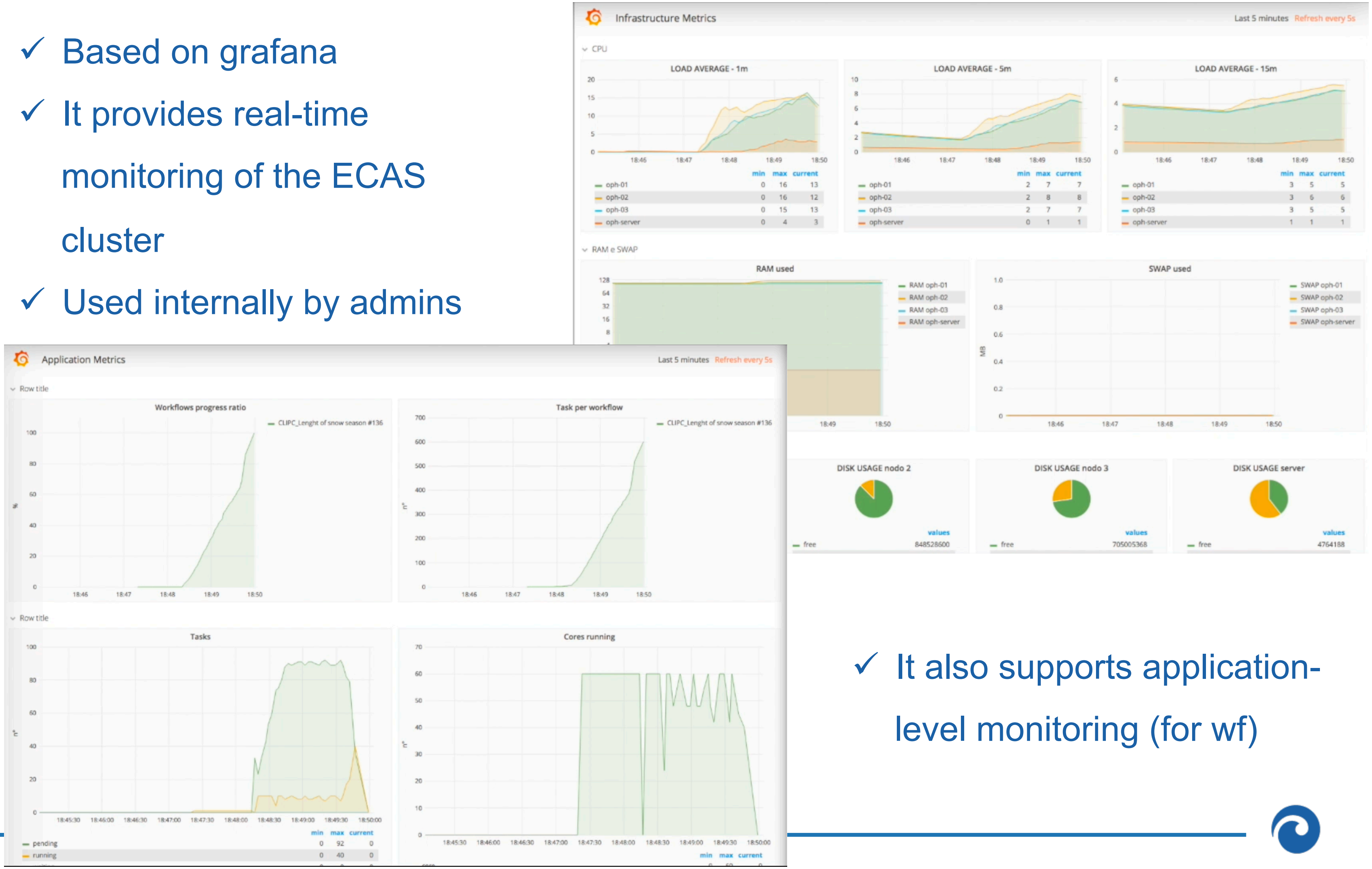
Dimension Information

NAME	TYPE	SIZE	HIERARCHY	CONCEPT LEVEL	ARRAY	LEVEL	LATTICE NAME
lat	double	1	oph_base	cell	no	1	
lon	double	1	oph_base	cell	no	2	
time	double	24	oph_base	cell	yes	1	

```
Execution time: 0.12 seconds
[40..1913] >>
```

ECASLab: Grafana monitoring interface

- ✓ Based on grafana
- ✓ It provides real-time monitoring of the ECAS cluster
- ✓ Used internally by admins

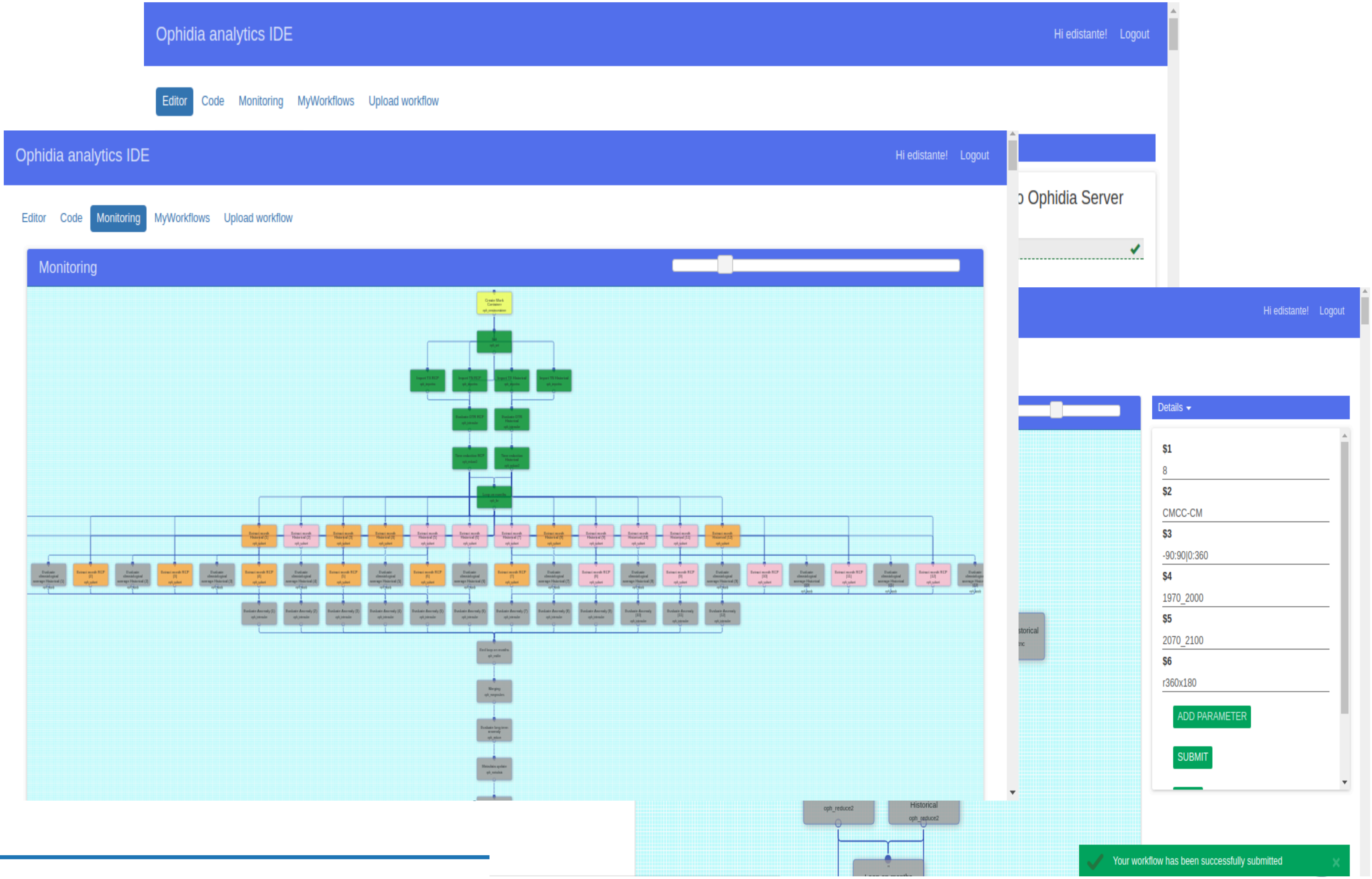


- ✓ It also supports application-level monitoring (for wf)

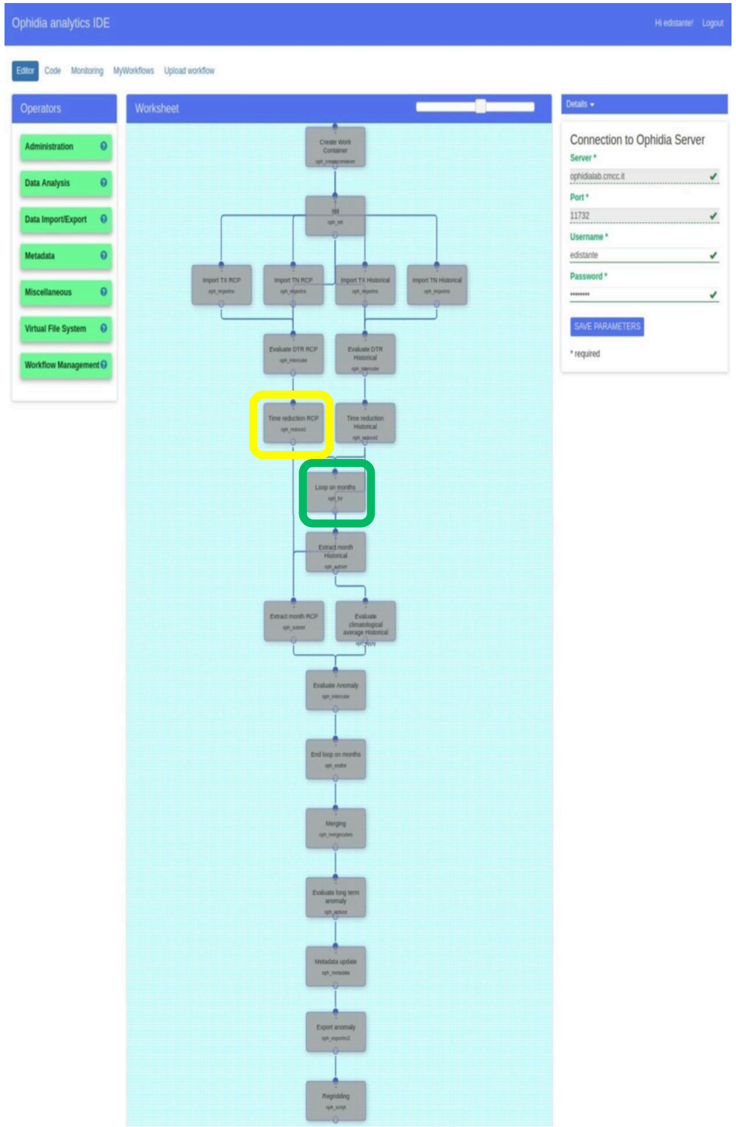
Looking forward

Workflow IDE and Server-side machine learning

ECASLab and the analytics workflow IDE



Easy and automated generation of JSON code




```
{
        "name": "dtr anomaly",
        "author": "CMCC",
        "abstract": "This workflow computes the anomaly of DTR (Diurnal Temperature Range) index with respect to past values. It works on two input files (tasmin/tasmax variable): $1 is ncores, $2 is the model, $3 is spatial filter (lat|lon ranges), $4 is the first time filter (historical), $5 is the second time filter (scenario), $6 is the grid of output map using the format r<lon>x<lat> (e.g. r360x180), i.e. a global regular lon/lat grid (this parameter is optional and by default the lon/lat grid of input file is adopted).",
        "exec_mode": "sync",
        "cwd": "/",
        "ncores": "${1}",
        "on_exit": "oph_delete",
        "tasks":
        [
                {
                        "name": "Init",
                        "operator": "oph_set",
                        "arguments": [
                                "key=missingvalue",
                                "value=1.e+20"
                        ]
                },
                {
                        "name": "Create Work Contai
                        "operator": "oph_createcont
                        "arguments": [
                                "container=dtr",
                                "dim=lat|lon|time",
                                "dim_type=double|do
                                "hierarchy=oph_base
                                "compressed=no",
                                "ncores=1",
                                "base_time=1850-01-
                                "calendar=standard"
                                "units=s"
                        ],
                        "on_error": "skip"
                },
                {
                        "name": "Import TX RCP",
                        "operator": "oph_importnc",
                        "arguments":
                        [
                                "src_path=/data/cmi
                                "measure=tasmax",
                                "base_time=1850-01-
                                "units=d",
```

OPH_REDUCE2

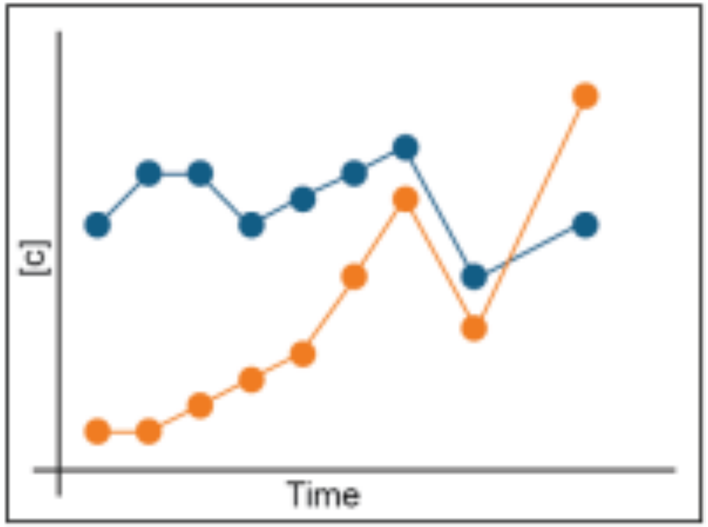
```
        "name": "Time reduction RCP",
        "operator": "oph_reduce2",
        "arguments":
        [
                "operation=avg",
                "dim=time",
                "concept_level=M",
                "description=Monthly Averages RCP",
                "missingvalue=@{missingvalue}"
        ],
        "dependencies": [
                { "task": "Evaluate DTR RCP", "type": "single" }
        ]
},
{
        "name": "Time reduction Historical",
        "operator": "oph_reduce2",
        "arguments":
        [
                "operation=avg",
                "dim=time",
                "concept_level=M",
                "description=Monthly Averages Historical",
                "missingvalue=@{missingvalue}"
        ],
        "dependencies": [
                { "task": "Evaluate DTR Historical", "type": "single" }
        ]
},
{
```

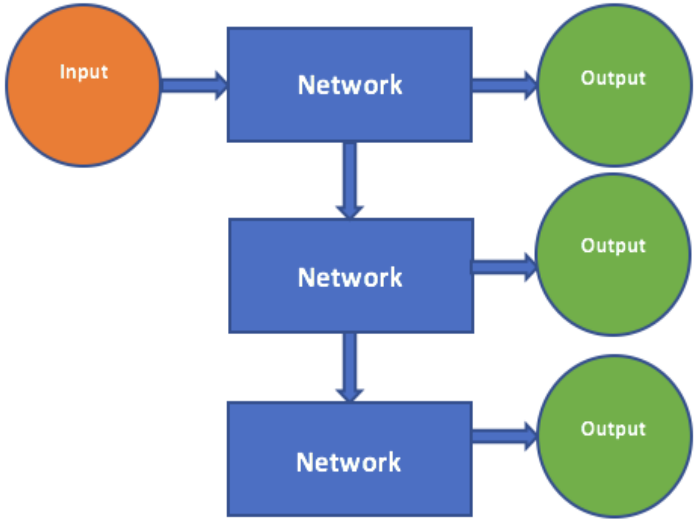
OPH_FOR

```
        "name": "Loop on months",
        "operator": "oph_for",
        "arguments": [
                "key=month",
                "counter=1:12",
                "parallel=yes"
        ],
        "dependencies": [
                { "task": "Time reduction RCP" },
                { "task": "Time reduction Historical" }
        ]
},
```

Long Short-Term Memory Network for Time Series Prediction

- We modeled the time series as a supervised learning problem, that is, as a sequence of inputs and outputs.
- At each stage, the network receives as input the *n* values in the past from a time *t*. The output is *h* nodes representing the values in the future.
- The goal of the network is to learn the mapping from the input to the output.
- Hopefully, the LSTM is able to capture some kind of temporal dependence in order to get better predictions.





Ophidia Primitives For LSTM: Training

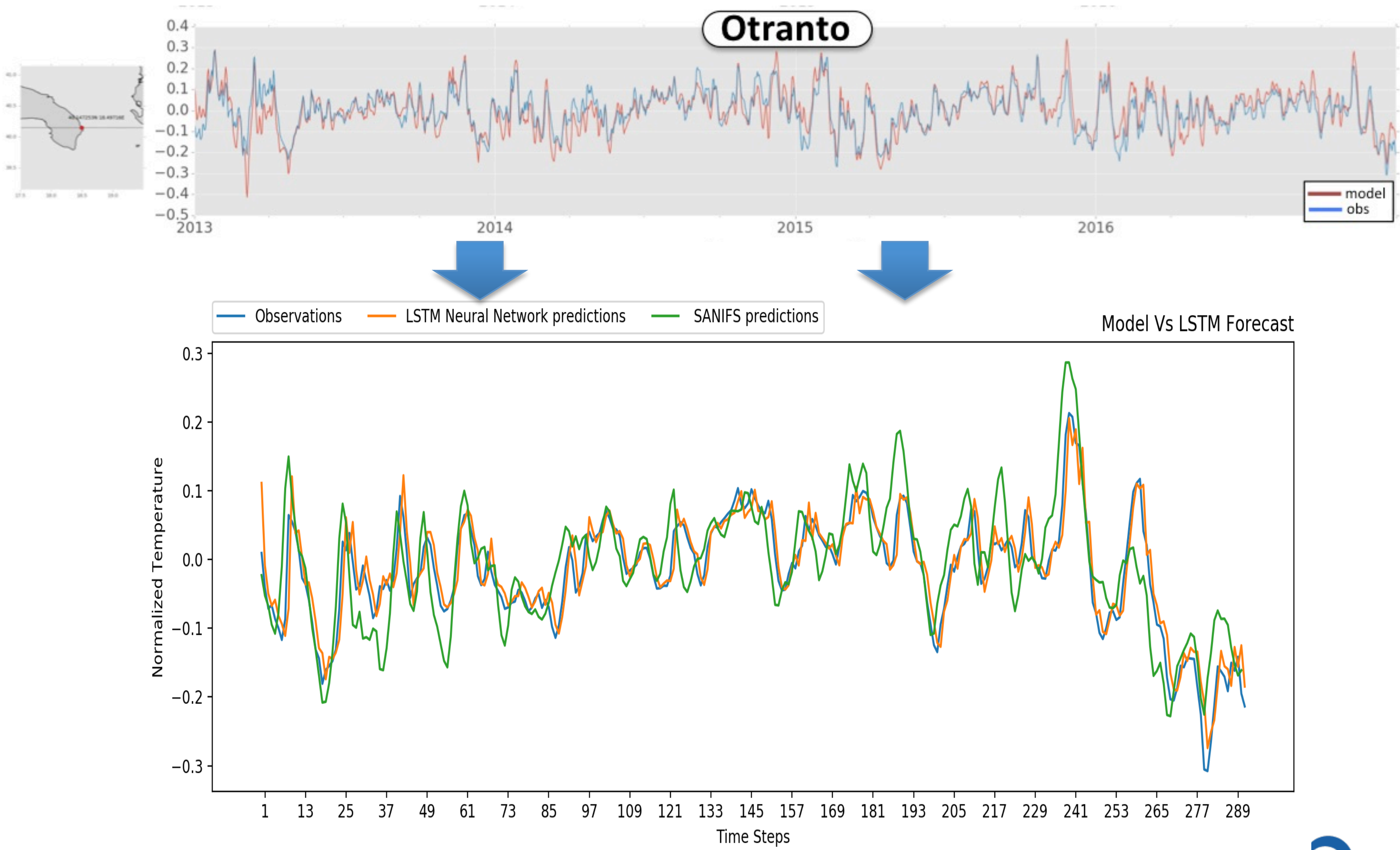
- The algorithm has been divided in two phases: one for training and one for test/prediction.
- The primitive for the **training** task:

```
oph_lstm(input_OPH_TYPE, output_OPH_TYPE, measure,
dim_in, dim_out, n_h_layers, n_h_neurons, [dropout],
[learning_rate], [unrolled_len], [minibatch_size],
[max_epoch])
```

- It can be run in a SQL statement or in the OPH_APPLY operator.
- After the training phase, the resulting neural network with updated parameters is saved as a binary array in a datacube. It can then be reused in the test phase.
- The primitive for the **test/prediction**:

```
oph_lstm_predict(input_OPH_TYPE, output_OPH_TYPE,
measure_a, measure_b, test)
```

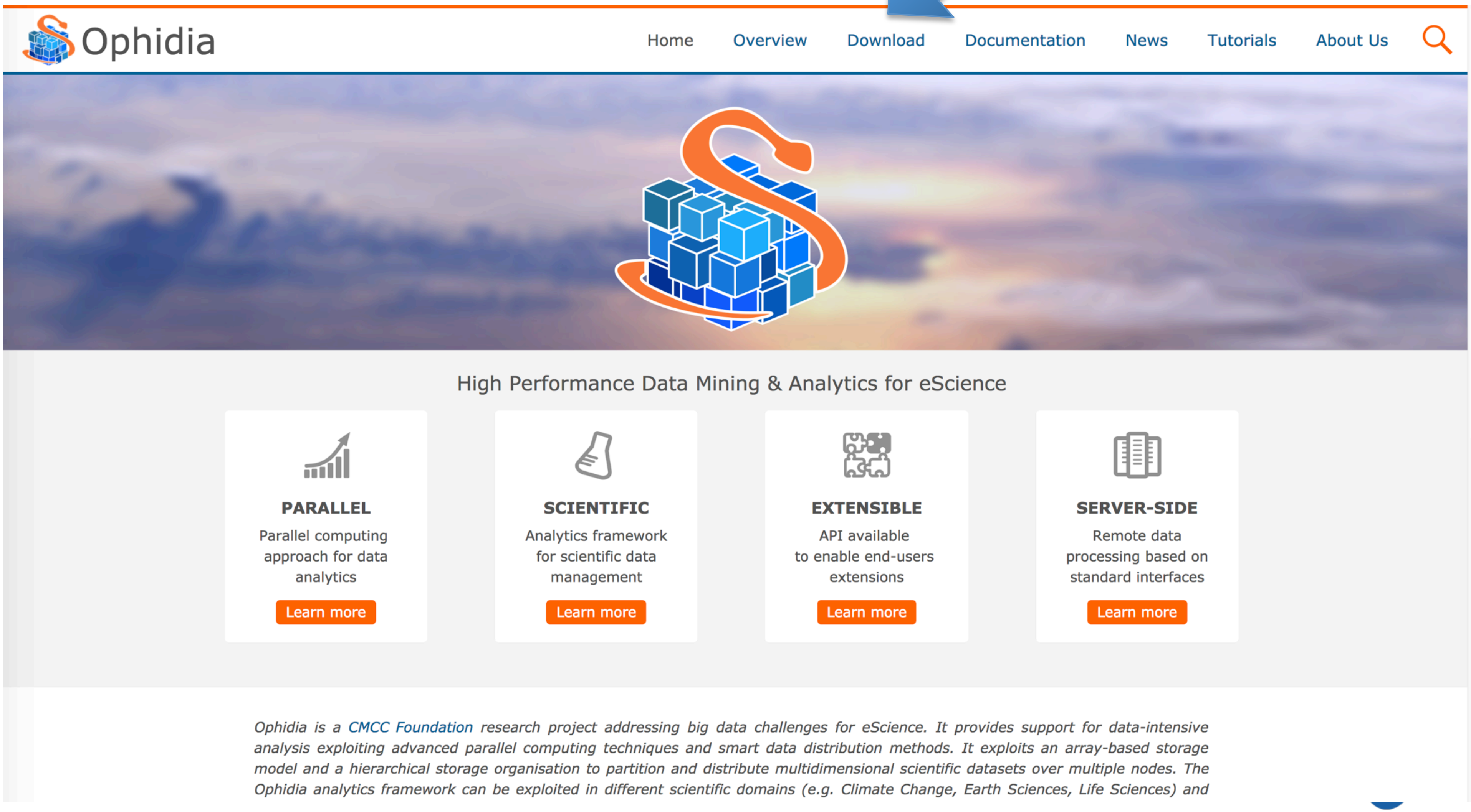
LSTM for the SANIFS Use Case



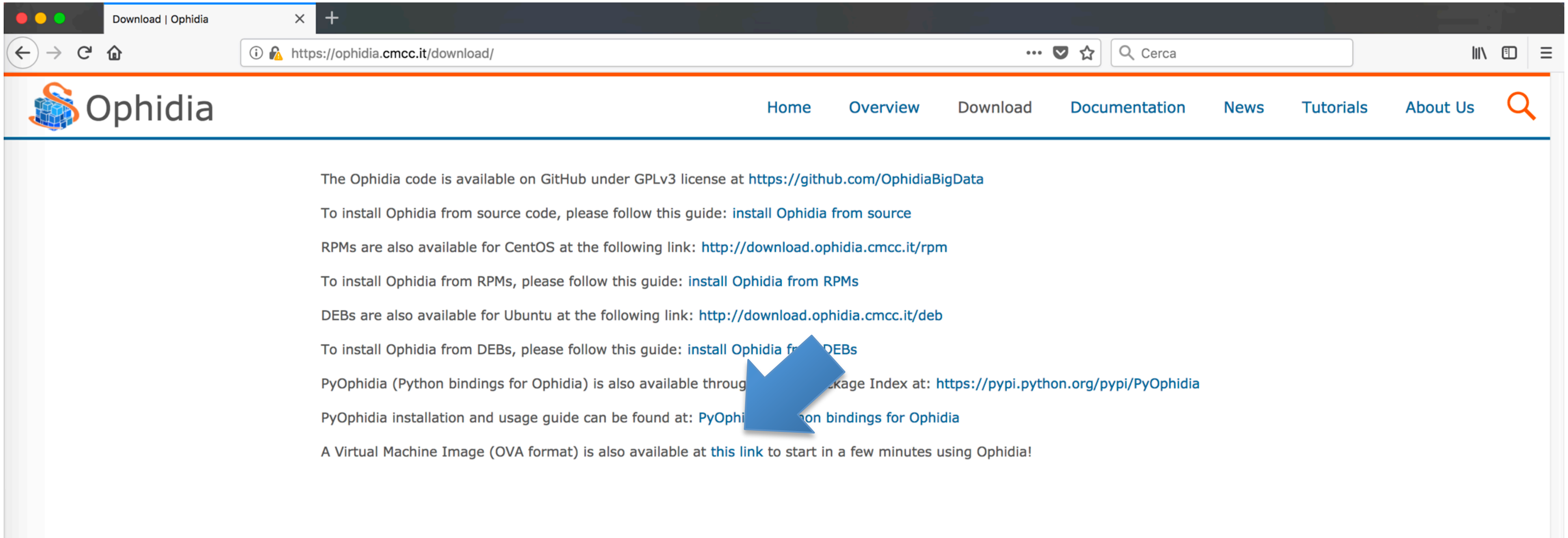
Useful resources and final remarks

Hands-on session – Quick Start

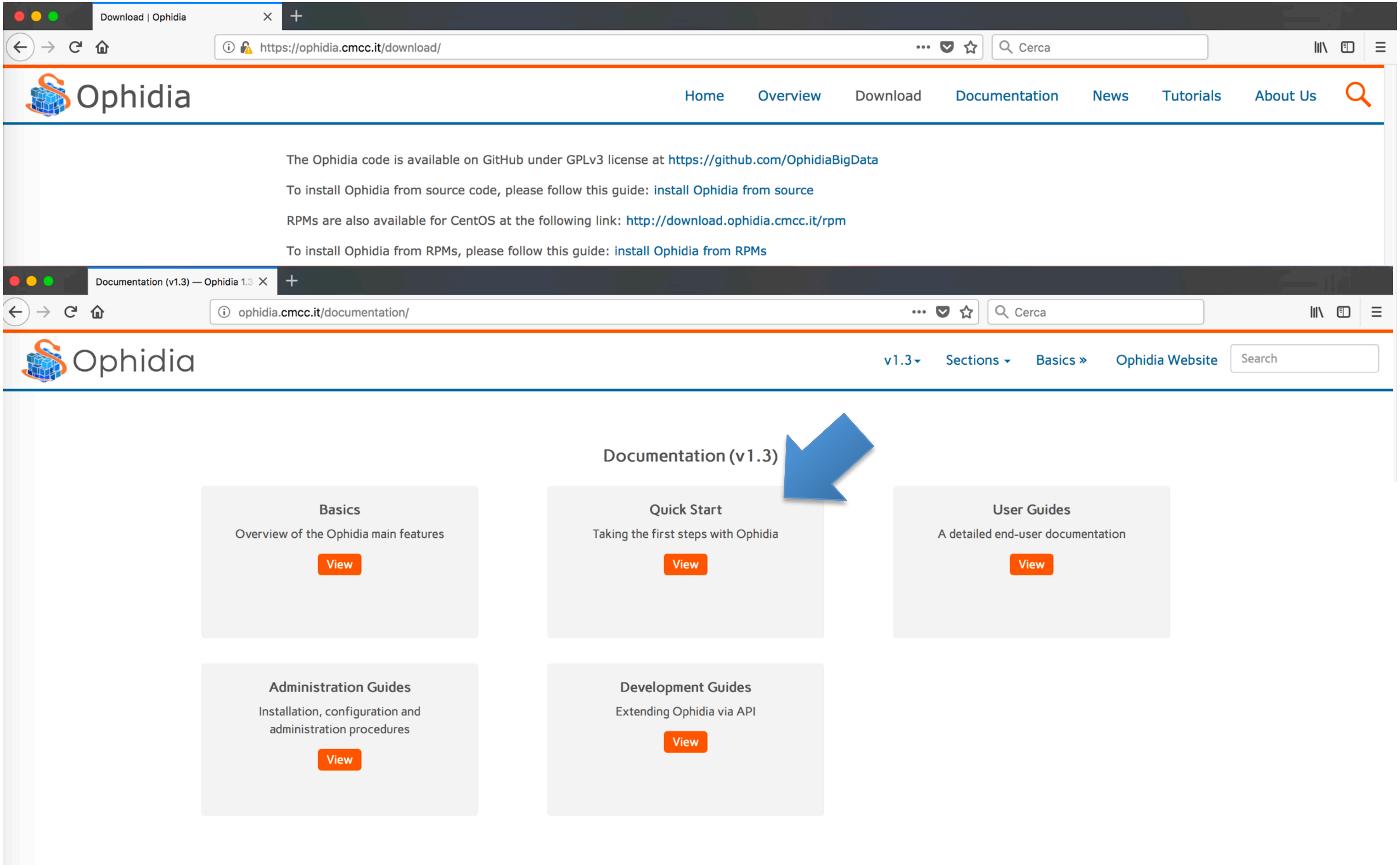
Website: http://ophidia.cmcc.it



Hands-on session – Quick Start



Hands-on session – Quick Start



Hands-on session – Quick Start

Ophidia Quick Start — Ophidia 1.3

ophidia.cmcc.it/documentation/general/quickstart.html

Cerca

Ophidia

v1.3 · Sections · « Basics · Ophidia User Guides » · Ophidia Website · Search

- Basics
- Quick Start
- User Guides
- Administration Guides
- Development Guides

Ophidia Quick Start

Let's start downloading the OphidiaVM (v1.2). The VMI for v1.3 will soon be available.

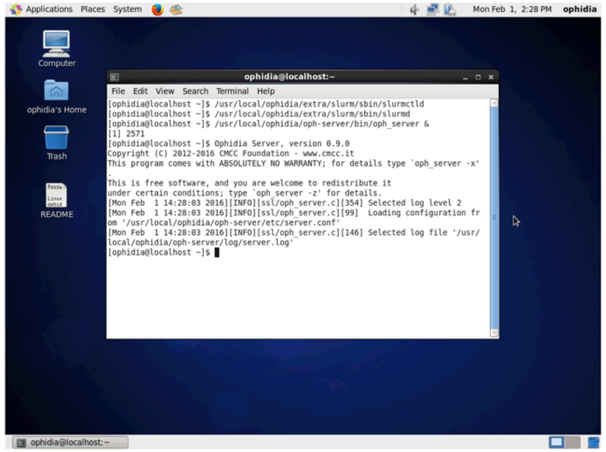
Import it inside the Virtual Box environment and start it.

Use the following credentials to login:

```
username: ophidia
password: ophidia
```

As *ophidia* user start the Ophidia Server using the command

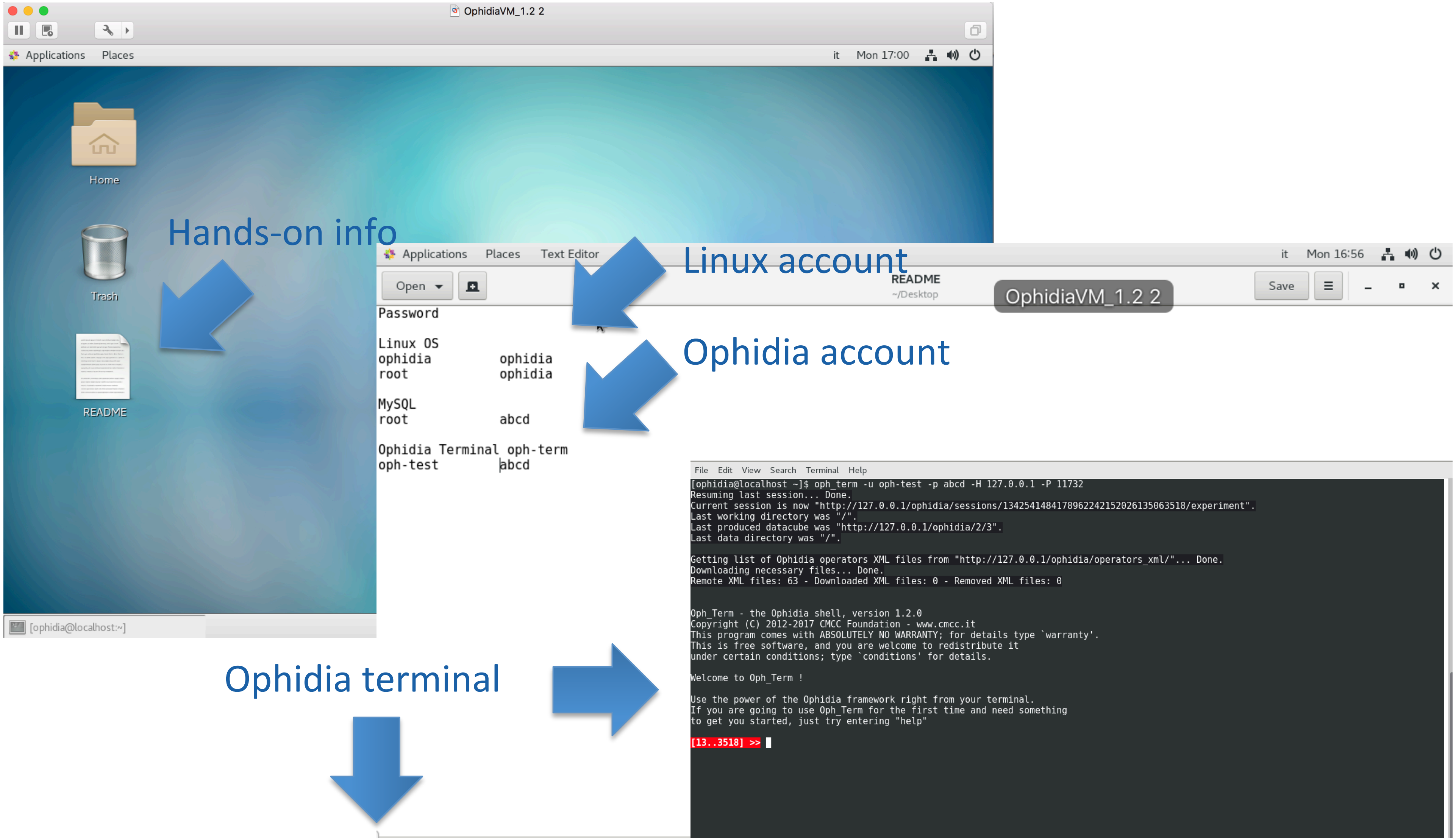
```
/usr/local/ophidia/oph-server/bin/oph_server &
```



Now you are ready to run the Ophidia terminal.

```
oph_term -u <user> -p <password> -H <Ophidia_Server_IP> -P <Ophidia_server_port>

The default parameters are:
```

Hands-on session – Accounts on the VM




```
[ophidia@localhost ~]$ oph_term -u oph-test -p abcd -H 127.0.0.1 -P 11732
```

Hands-on session – Jupyter-Hub

Website: http://ophidialab.cmcc.it

Ophidia

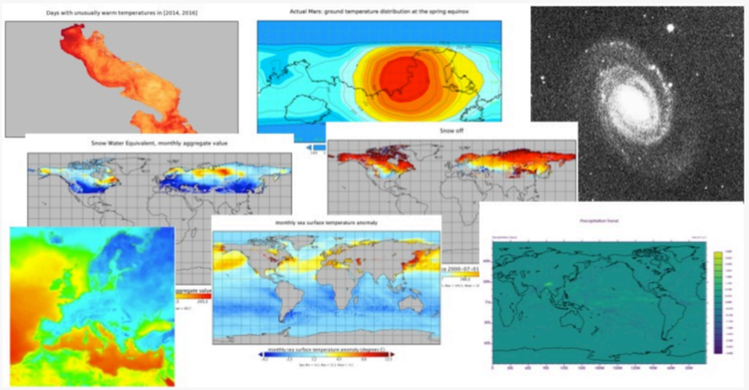
Home Quick Start JupyterHub Experiments Monitoring Support Register

ECASLab

ECASLab is a scientific data analytics environment. It builds on top of ECAS (the ENES Climate Analytics Service), one of the thematic services included in the EOSC-hub service portfolio.
ECASLab starts from a previous effort (OphidiaLab, developed at CMCC Foundation) with the main aim of providing a virtualized research environment for researchers. It represents the entry point for users that want to test, train, exploit the ECAS Thematic Service.

ECASLab provides a scientific environment exploiting a server-side approach and integrating both data and analysis tools to support data scientists in their daily research activities.
It consists of several components like an ECAS cluster, a JupyterHub instance jointly with a large set of pre-installed Python libraries for running data manipulation, analysis, and visualization, a data publication service and a tool for the infrastructure monitoring (mainly intended for the administrators).

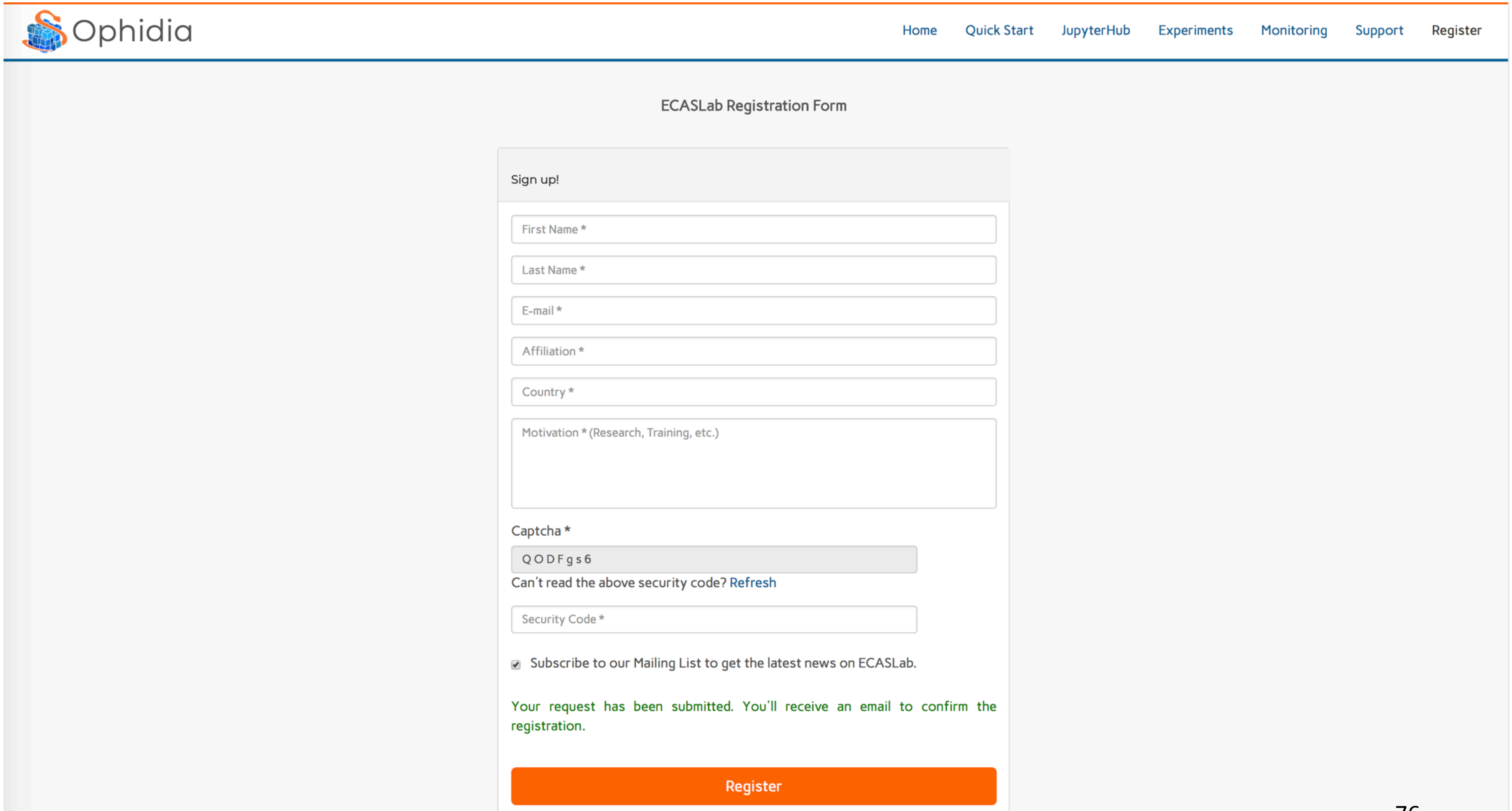
In order to get started with ECASLab please have a look at the Quick Start guide and register here to get an account.



A few examples of output related to different analytics experiments implemented in the ECASLab environment.

Hands-on session – Jupyter-Hub

Website: http://ophidialab.cmcc.it


Ophidia

Home | Quick Start | JupyterHub | Experiments | Monitoring | Support | Register

ECASLab Registration Form

Sign up!

First Name *

Last Name *

E-mail *

Affiliation *

Country *

Motivation * (Research, Training, etc.)

Captcha *

QODFgs6

Can't read the above security code? Refresh

Security Code *

Subscribe to our Mailing List to get the latest news on ECASLab.

Your request has been submitted. You'll receive an email to confirm the registration.

Register

Hands-on session – Jupyter-Hub

Website: http://ophidialab.cmcc.it

Ophidia

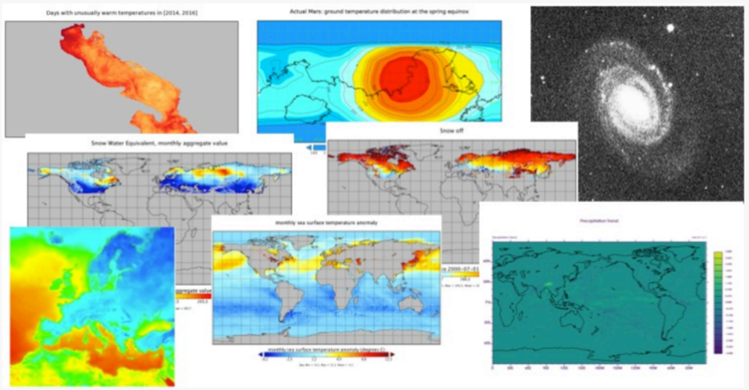
Home Quick Start JupyterHub Experiments Monitoring Support Register

ECASLab

ECASLab is a scientific data analytics environment. It builds on top of ECAS (the ENES Climate Analytics Service), one of the thematic services included in the EOSC-hub service portfolio.
ECASLab starts from a previous effort (OphidiaLab, developed at CMCC Foundation) with the main aim of providing a virtualized research environment for researchers. It represents the entry point for users that want to test, train, exploit the ECAS Thematic Service.

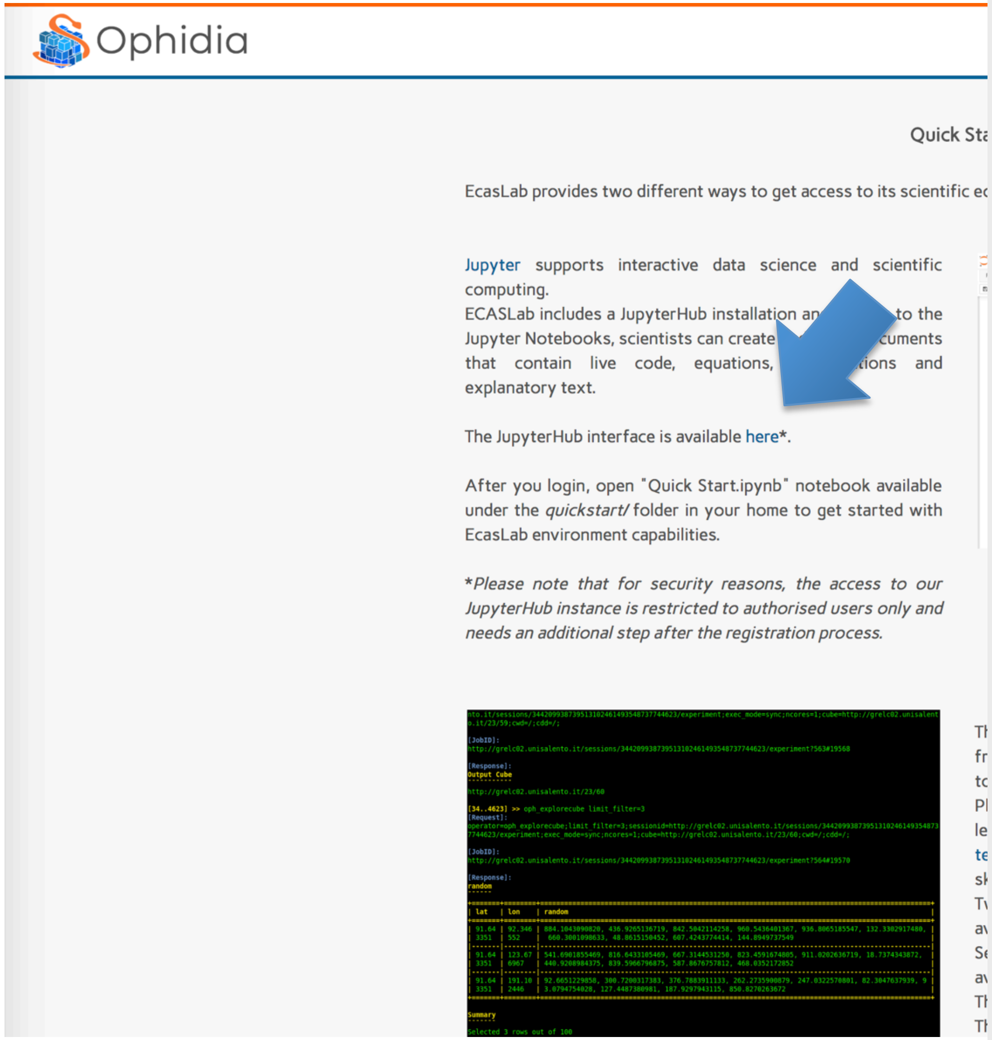
ECASLab provides a scientific environment exploiting a server-side approach and integrating both data and analysis tools to support data scientists in their daily research activities.
It consists of several components like an ECAS cluster, a JupyterHub instance jointly with a large set of pre-installed Python libraries for running data manipulation, analysis, and visualization, a data publication service and a tool for the infrastructure monitoring (mainly intended for the administrators).

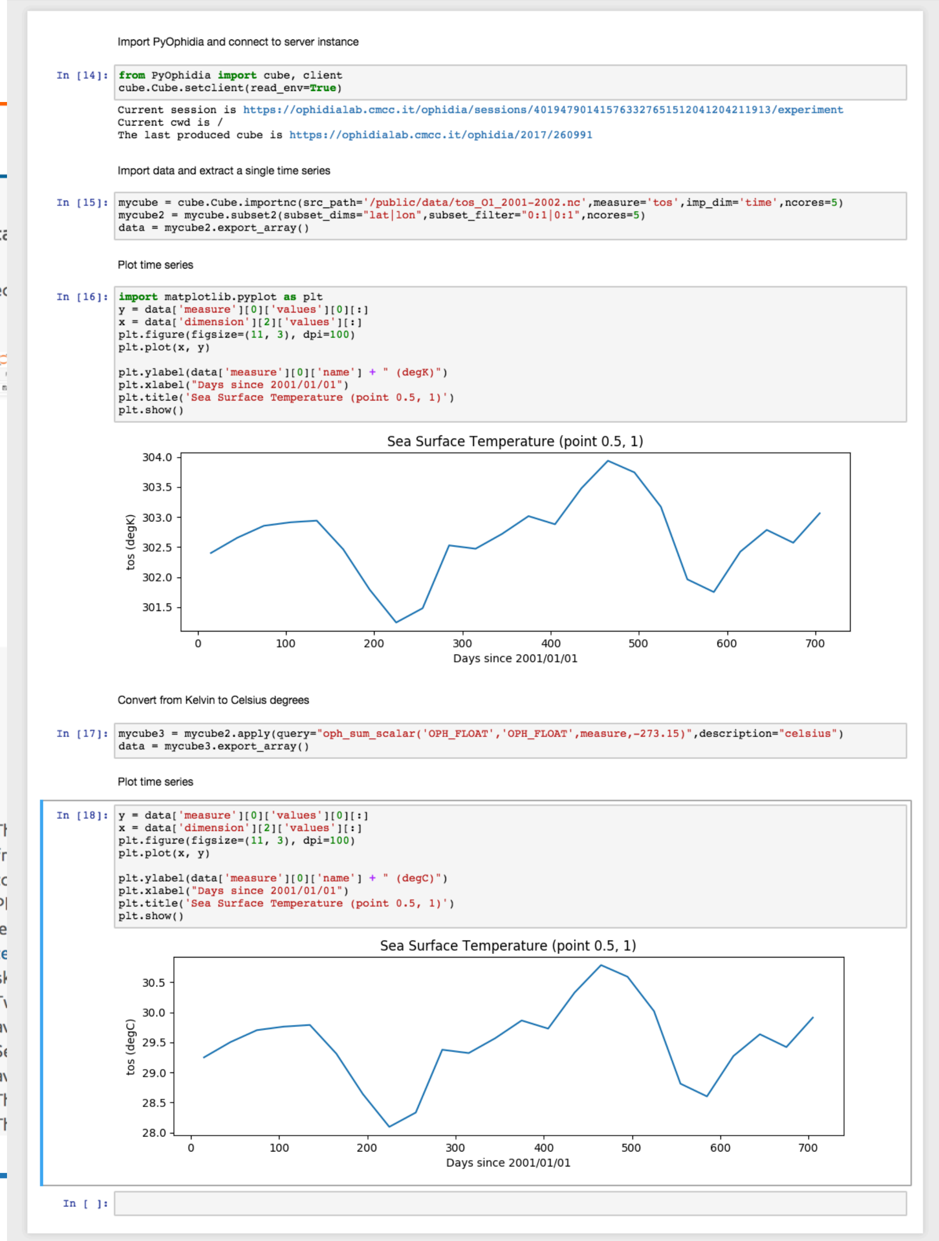
In order to get started with ECASLab please have a look at the Quick Start guide and register here to get an account.



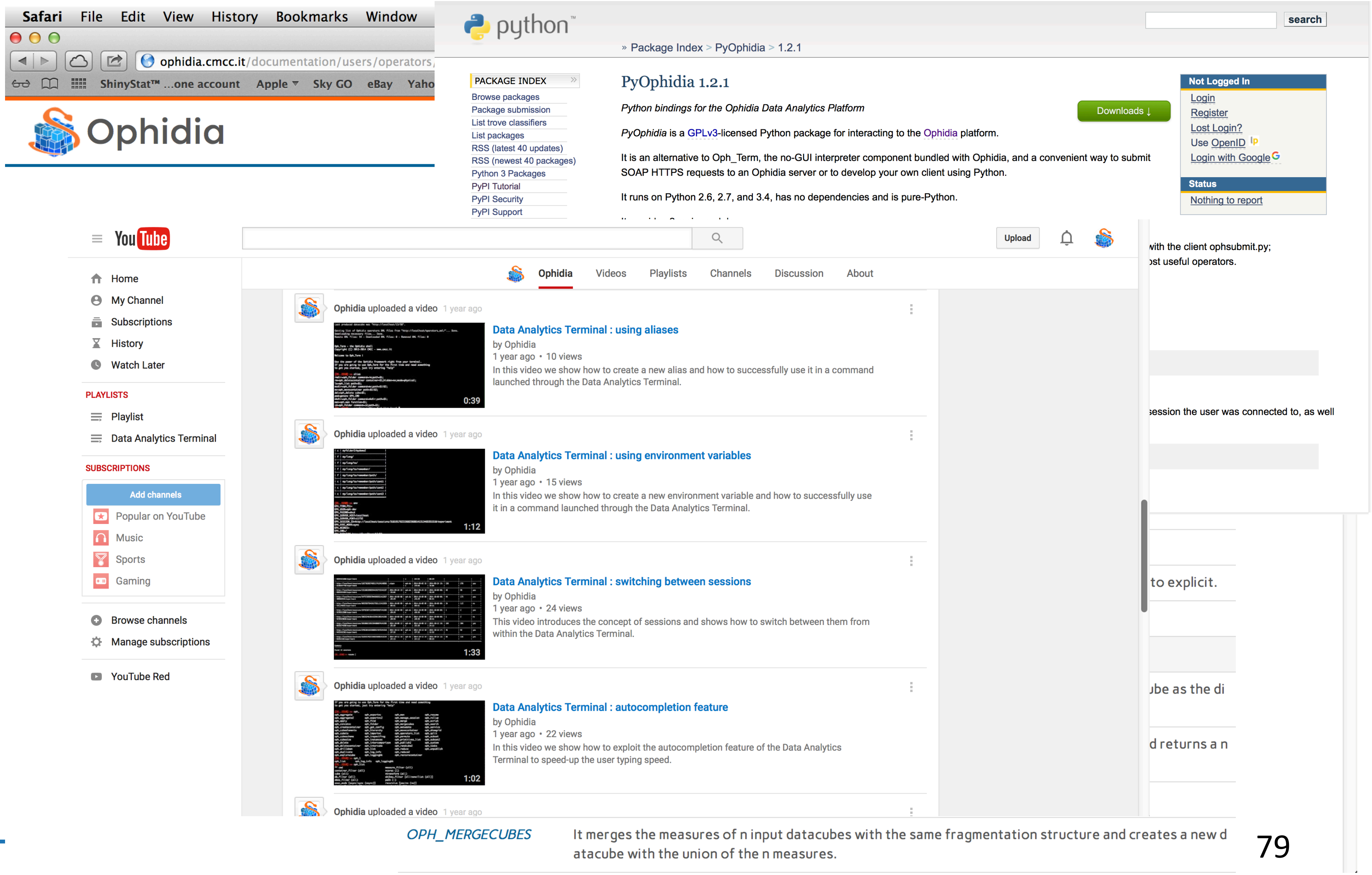
A few examples of output related to different analytics experiments implemented in the ECASLab environment.

Hands-on session – Jupyter-Hub





Ophidia documentation and social/multimedia content



Useful Resources

- *Website: https://ophidia.cmcc.it*
- *Doc : http://ophidia.cmcc.it/documentation*
- *The Ophidia code is available on GitHub under GPLv3 license at https://github.com/OphidiaBigData*
- *RPMs are also available for CentOS6 at the following repo: http://download.ophidia.cmcc.it/rpm*
- *Youtube Channel*
  https://www.youtube.com/user/OphidiaBigData/
- *A Virtual Machine Image (OVA format) is also available at https://download.ophidia.cmcc.it/vmi_desktop/ to get started in a few minutes with Ophidia*

Publications

[11] S. Fiore, C. Palazzo, A. D'Anca, D. Elia, E. Londero, C. Knapic, S. Monna, N. M. Marcucci, F. Aguilar, M. Płóciennik, J. E. M. De Lucas, G. Aloisio, "Big Data Analytics on Large-Scale Scientific Datasets in the INDIGO-DataCloud Project". In Proceedings of the ACM International Conference on Computing Frontiers (CF '17), May 15-17, 2017, Siena, Italy, pp. 343-348

[10] A. D'Anca, C. Palazzo, D. Elia, S. Fiore, I. Bistinas, K. Böttcher, V. Bennett, G. Aloisio, "On the Use of In-memory Analytics Workflows to Compute eScience Indicators from Large Climate Datasets," 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, May 14-17, 2017, pp. 1035-1043.

[9] S. Fiore, M. Plóciennik, C. M. Doutriaux, C. Palazzo, J. Boutte, T. Zok, D. Elia, M. Owsiak, A. D'Anca, Z. Shaheen, R. Bruno, M. Fargetta, M. Caballer, G. Moltó, I. Blanquer, R. Barbera, M. David, G. Donvito, D. N. Williams, V. Anantharaj, D. Salomoni, G. Aloisio, "Distributed and cloud-based multi-model analytics experiments on large volumes of climate change data in the earth system grid federation eco-system". In Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016. p. 2911-2918.

[8] M. Plociennik, S. Fiore, G. Donvito, M. Owsiak, M. Fargetta, R. Barbera, R. Bruno, E. Giorgio, D. N. Williams, and G. Aloisio, "Two-level Dynamic Workflow Orchestration in the INDIGO DataCloud for Large-scale, Climate Change Data Analytics Experiments", International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA. Procedia Computer Science, vol. 80, 2016, pp. 722-733

[7] D. Elia, S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. N. Williams, G. Aloisio, "An in-memory based framework for scientific data analytics". In Proceedings of the ACM International Conference on Computing Frontiers (CF '16), May 16-19, 2016, Como, Italy, pp. 424-429

[6] C. Palazzo, A. Mariello, S. Fiore, A. D'Anca, D. Elia, D. N. Williams, G. Aloisio, "A Workflow-Enabled Big Data Analytics Software Stack for eScience", The Second International Symposium on Big Data Principles, Architectures & Applications (BDAA 2015), HPCS 2015, Amsterdam, The Netherlands, July 20-24, 2015, pp. 545-552

[5] S. Fiore, M. Mancini, D. Elia, P. Nassisi, F. V. Brasileiro, I. Blanquer, I. A. A. Rufino, A.C. Seijmonsbergen, C. O. Galvao, V. P. Canhos, A. Mariello, C. Palazzo, A. Nuzzo, A. D'Anca, G. Aloisio, "Big data analytics for climate change and biodiversity in the EUBrazilCC federated cloud infrastructure", Workshop on Analytics Platforms for the Cloud, In Proceedings of the 12th ACM International Conference on Computing Frontiers (CF '15), May 18th, 2015, Ischia, Italy. Article 52, 8 pages.

[4] S. Fiore, A. D'Anca, D. Elia, C. Palazzo, I. Foster, D. Williams, G. Aloisio, "Ophidia: A Full Software Stack for Scientific Data Analytics", proc. of the 2014 International Conference on High Performance Computing & Simulation (HPCS 2014), July 21 – 25, 2014, Bologna, Italy, pp. 343-350, ISBN: 978-1-4799-5311-0

[3] S. Fiore, C. Palazzo, A. D'Anca, I. T. Foster, D. N. Williams, G. Aloisio, "A big data analytics framework for scientific data management", IEEE BigData Conference 2013: 1-8

[2] S. Fiore, A. D'Anca, C. Palazzo, I. T. Foster, D. N. Williams, G. Aloisio, "Ophidia: Toward Big Data Analytics for eScience", ICCS 2013, June 5-7, 2013 Barcelona, Spain, ICCS, volume 18 of Procedia Computer Science, page 2376-2385. Elsevier, 2013

[1] G. Aloisio, S. Fiore, I. Foster, D. Williams , "Scientific big data analytics challenges at large scale", Big Data and Extreme-scale Computing (BDEC), April 30 to May 01, 2013, Charleston, South Carolina, USA (position paper).

Conclusions

- ✔ ***ECAS*** *represents the community evolution of Ophidia and is a key thematic service in the context of the* ***EOSC-hub***
- ✔ ***OLAP approach*** *for big data – multidimensional data model*
- ✔ *Multiple use cases for data analysis in* ***different domains*** *have been implemented*
- ✔ *It provides access via* ***CLI*** *(end-users) and* ***API*** *(devel users)*
- ✔ *Programmatic access via* ***C*** *and* ***Python APIs***
- ✔ *Several deployment scenarios tested in* ***cloud*** *and* ***HPC*** *environments*
- ✔ *Strong* ***workflow support*** *and* ***in-memory analytics***
- ✔ ***ECASLab*** *integrates several* ***UNIDATA*** *software (****NetCDF lib, THREDDS , IDV****)*
- ✔ ***Official Release*** *available from February 1st 2016 on github*
  - ✔ *Latest Release* ***v1.3*** *released in June (last week)*

Do you want to join?

*That's an **open source** effort aiming at becoming a **community effort***

I'll be very happy to know what aspects of this project you are more interested in

Feel free to get in touch with us

sandro.fiore@cmcc.it

Thanks

http://ophidia.cmcc.it

@OphidiaBigData

www.youtube.com/user/OphidiaBigData

enes
EUROPEAN NETWORK
FOR EARTH SYSTEM MODELLING

EOSC-hub

EOSC-hub receives funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 777536.