High-Perfomance Data Analytics in eScience and the Ophidia Framework

D. Elia[1,2], **F. Antonio**[1]

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

ESiWACE2 online training course on High-Performance Data Analytics and Visualisation

1st session

6 October 2020

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

cmcc
Centro Euro-Mediterraneo sui Cambiamenti Climatici

Session outline

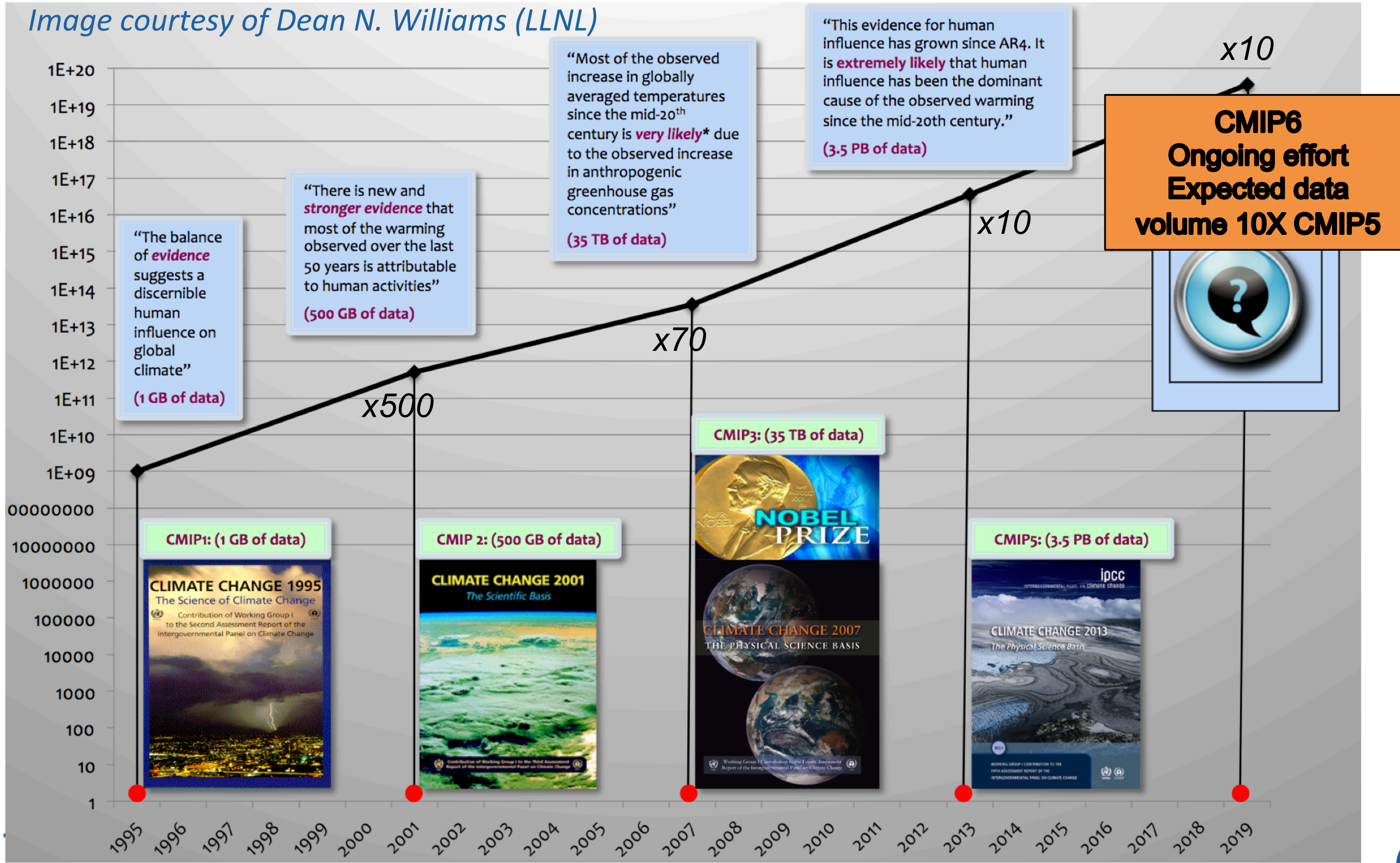
- ✔ *Introduction data challenges in eScience and HPDA*
- ✔ *Introduction to the Ophidia HPDA Framework*
- ✔ *Ophidia core concepts: architecture, data model, operators and primitives*
- ✔ *Ophidia Python bindings: PyOphidia*
- ✔ *DEMO: Introduction to PyOphidia*
- ✔ *HANDS-ON: Data analytics examples with PyOphidia*

Climate analysis challenges & issues

Several key challenges and practical issues related to large-scale climate analysis

- Setup of a data analysis experiment requires the ***download of (multiple) input data***
  - *Data download is a big barrier for climate scientists*
  - *Reducing data movement is essential*
- The complexity of the analysis leads to the need for ***end-to-end workflow support***
  - *Data analysis* mainly performed using *client-side* approaches
  - Analysing large datasets involves *running tens/hundreds of analytics operators*
- Large data volumes pose **strong requirements in terms of computational and storage resources**

CMIP data evolution



Convergence of data analytics and HPC in eScience

- *(Big) Data analytics ecosystem has rapidly expanded in the last 15 years, leading to a wide spectrum of new solutions, mainly outside the scientific and engineering community*
- *HPC solutions have been used for several years in different scientific fields for scientific computing (simulations and modeling)*
- *Computational science modeling and data analytics are both crucial in scientific research*
- *The convergence of the solutions and technology of the two ecosystems is a key factor for accelerating scientific discovery*

High-Performance Data Analytics (HPDA)

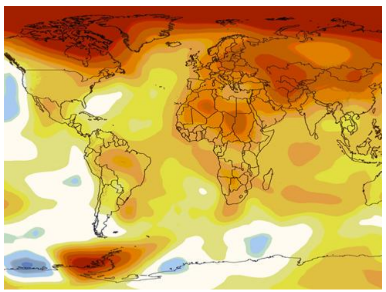
Ophidia High-Performance Data Analytics Framework

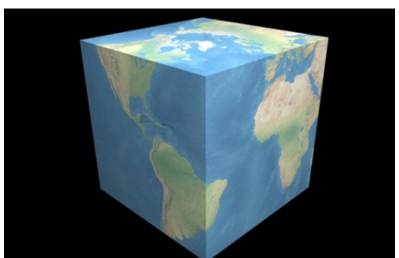
Ophidia (http://ophidia.cmcc.it) is a CMCC Foundation research project addressing data challenges for eScience

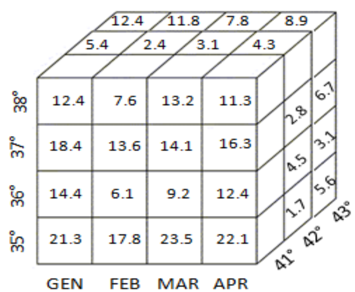
- a *High-Performance Data Analytics* (HPDA) framework for multi-dimensional scientific data joining HPC paradigms with scientific data analytics approaches
- in-memory and server-side data analysis exploiting parallel computing techniques and database approaches
- a multi-dimensional, array-based, storage model and partitioning schema for scientific data leveraging the datacube abstraction
- end-to-end mechanisms to support complex experiments and large workflows on scientific datacubes, primarily in climate domain

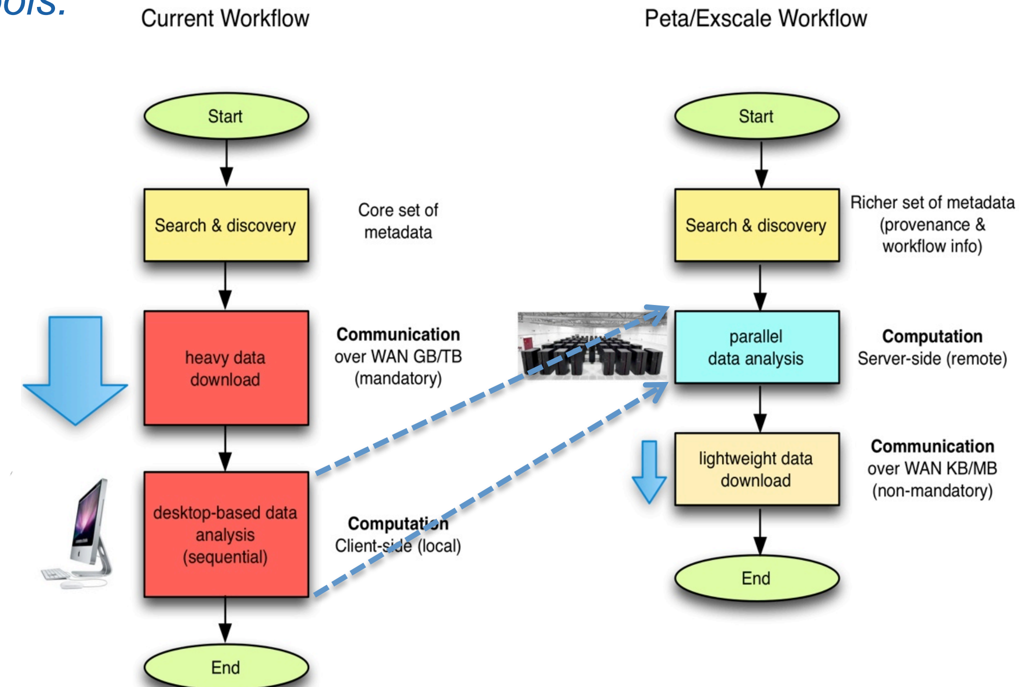
A paradigm shift

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.







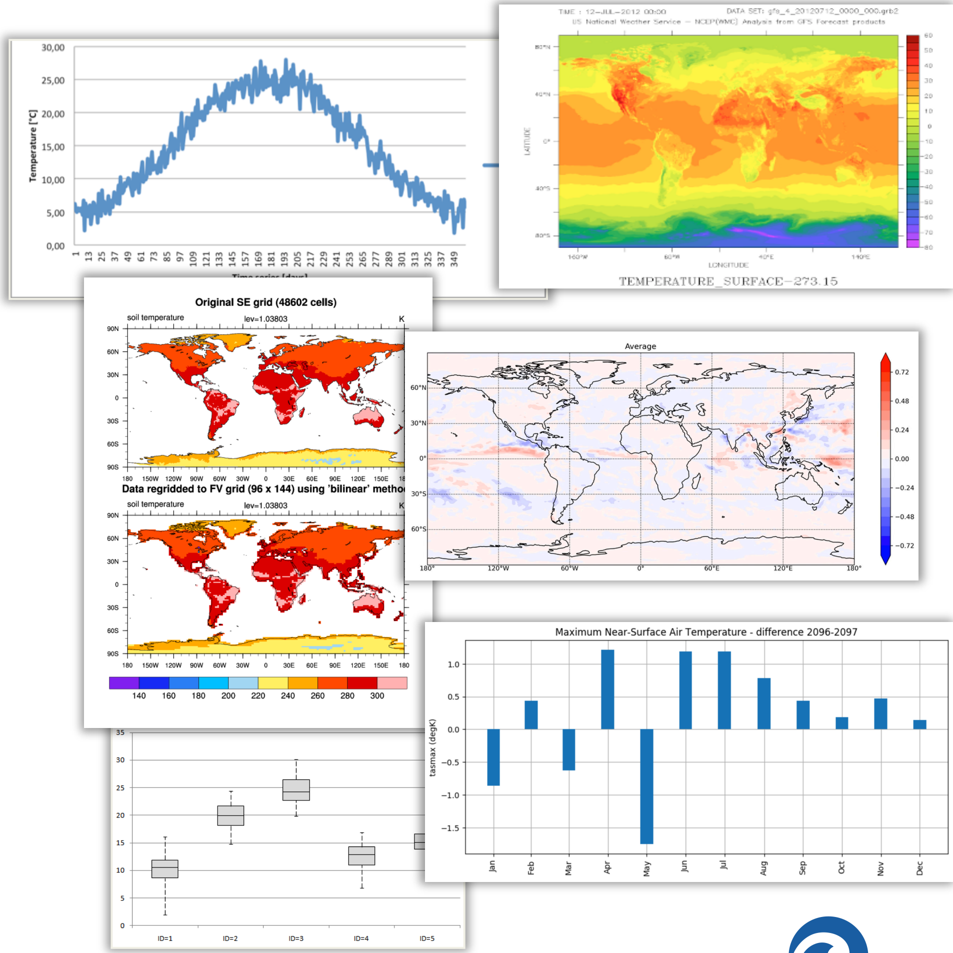


S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. N. Williams, G. Aloisio, "Ophidia: toward bigdata analytics for eScience", ICCS2013 Conference, Procedia Elsevier, Barcelona, June 5-7, 2013

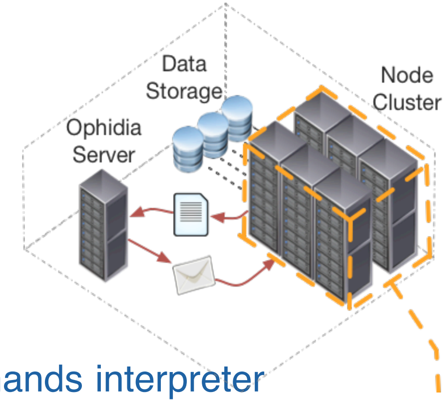
Data analytics requirements and use cases

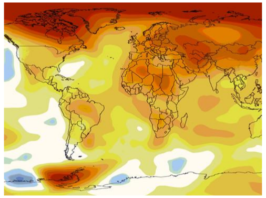
Requirements and needs focus on:

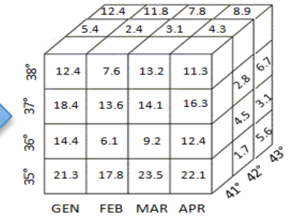
- *Time series analysis*
- *Data subsetting*
- *Model intercomparison*
- *Multi-model means*
- *Massive data reduction*
- *Data transformation*
- *Parameter sweep experiments*
- *Maps generation*
- *Ensemble analysis*
- *Data analytics workflow support*



Server-side paradigm and the datacube abstraction





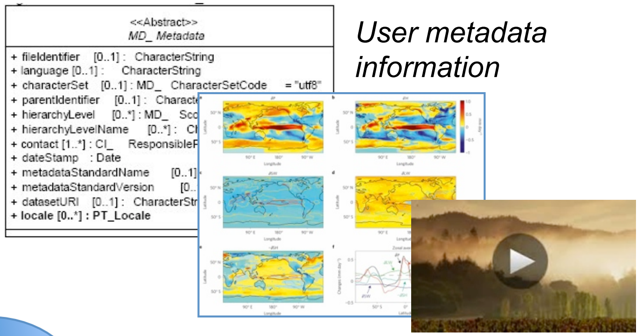


Oph_Term: a terminal-like commands interpreter serving as a client for the Ophidia framework

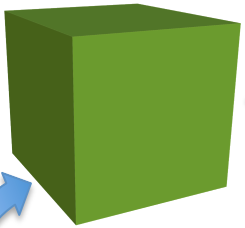
PyOphidia: a Python interface for datacube management & analytics with Ophidia

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

Through **oph_term/PyOphidia** the user run ("send") commands ("operators") to the Ophidia framework to manipulate datasets ("datacubes")



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)
```

Some international projects exploiting Ophidia

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

is-enes
INFRASTRUCTURE FOR THE EUROPEAN NETWORK FOR EARTH SYSTEM MODELLING

EOSC-hub

EUBRA BIGSEA
EUROPE - BRAZIL COLLABORATION OF BIG DATA SCIENTIFIC RESEARCH THROUGH CLOUD-CENTRIC APPLICATIONS

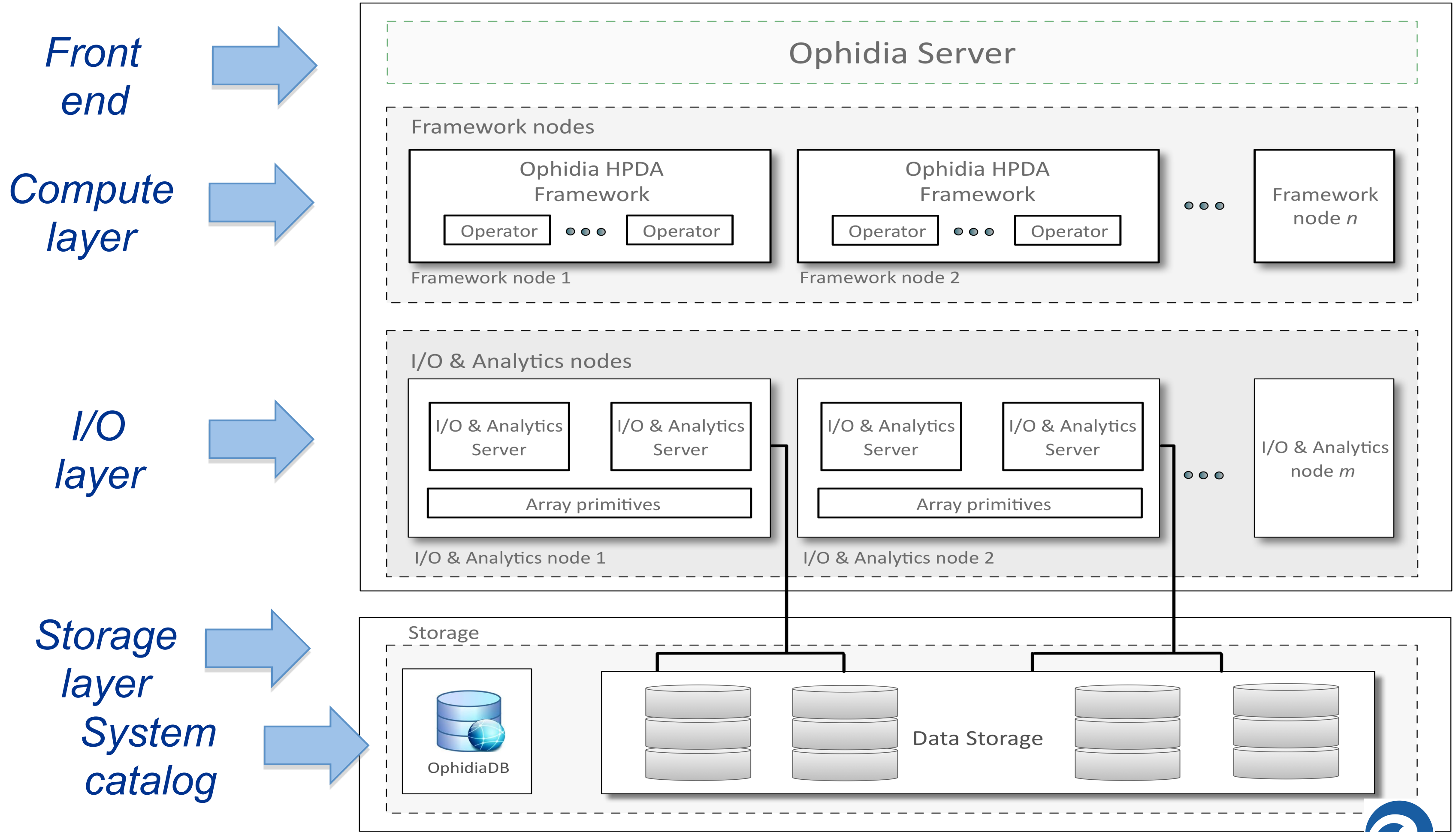
Clipc
Climate Information Portal

Ofidia

INDIGO - DataCloud

EU Brazil Cloud Connect
EU Brazil Cloud Computing for Science

Ophidia architecture: overview

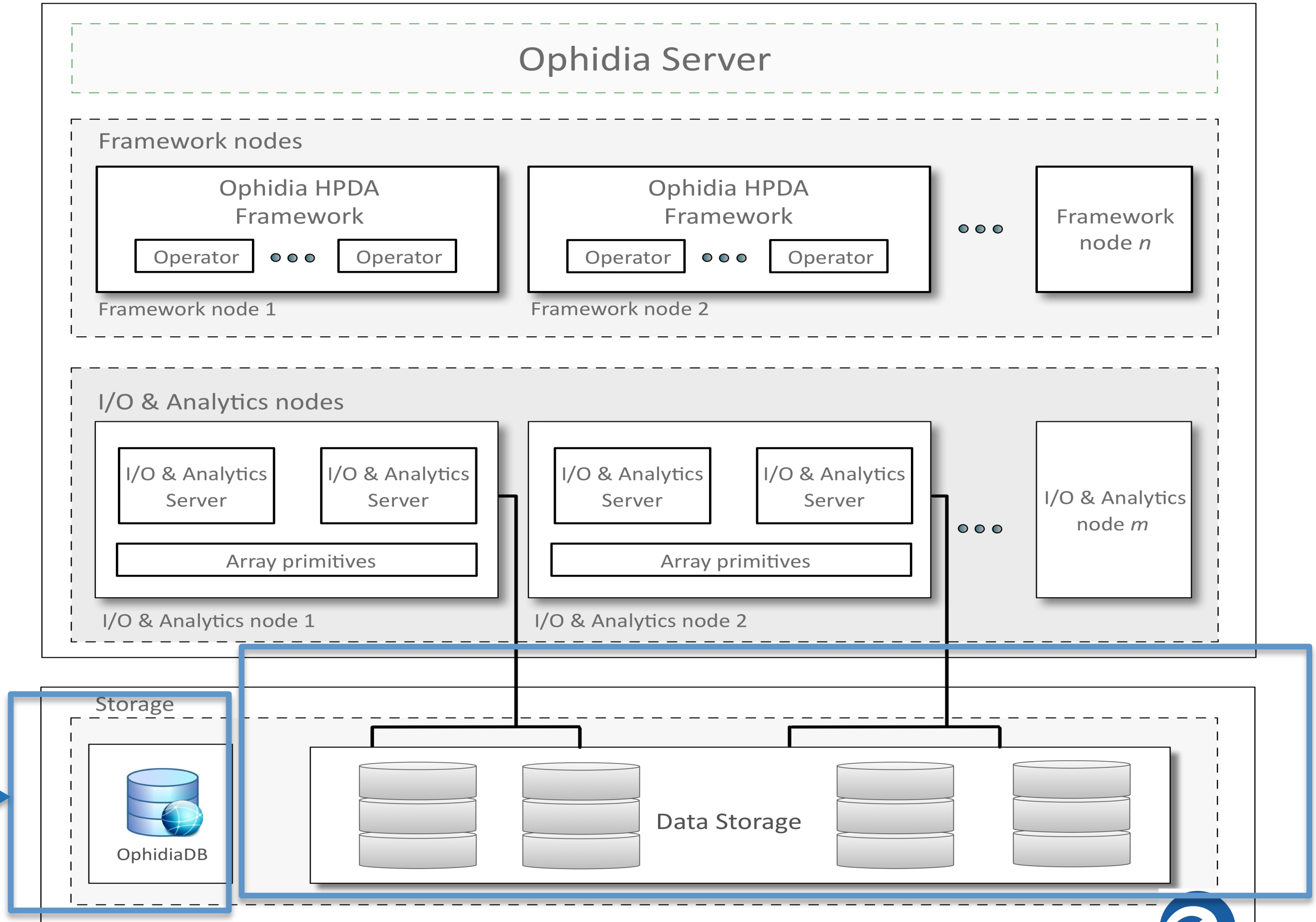


Ophidia architecture: storage layer

Distributed *hardware resources to manage storage*

Data partitioned *in a hierarchical fashion over the storage according to the storage model & partitioning schema*

OphidiaDB is the system catalog: maps data fragmentation and tracks metadata



Multi-dimensional storage model implementation

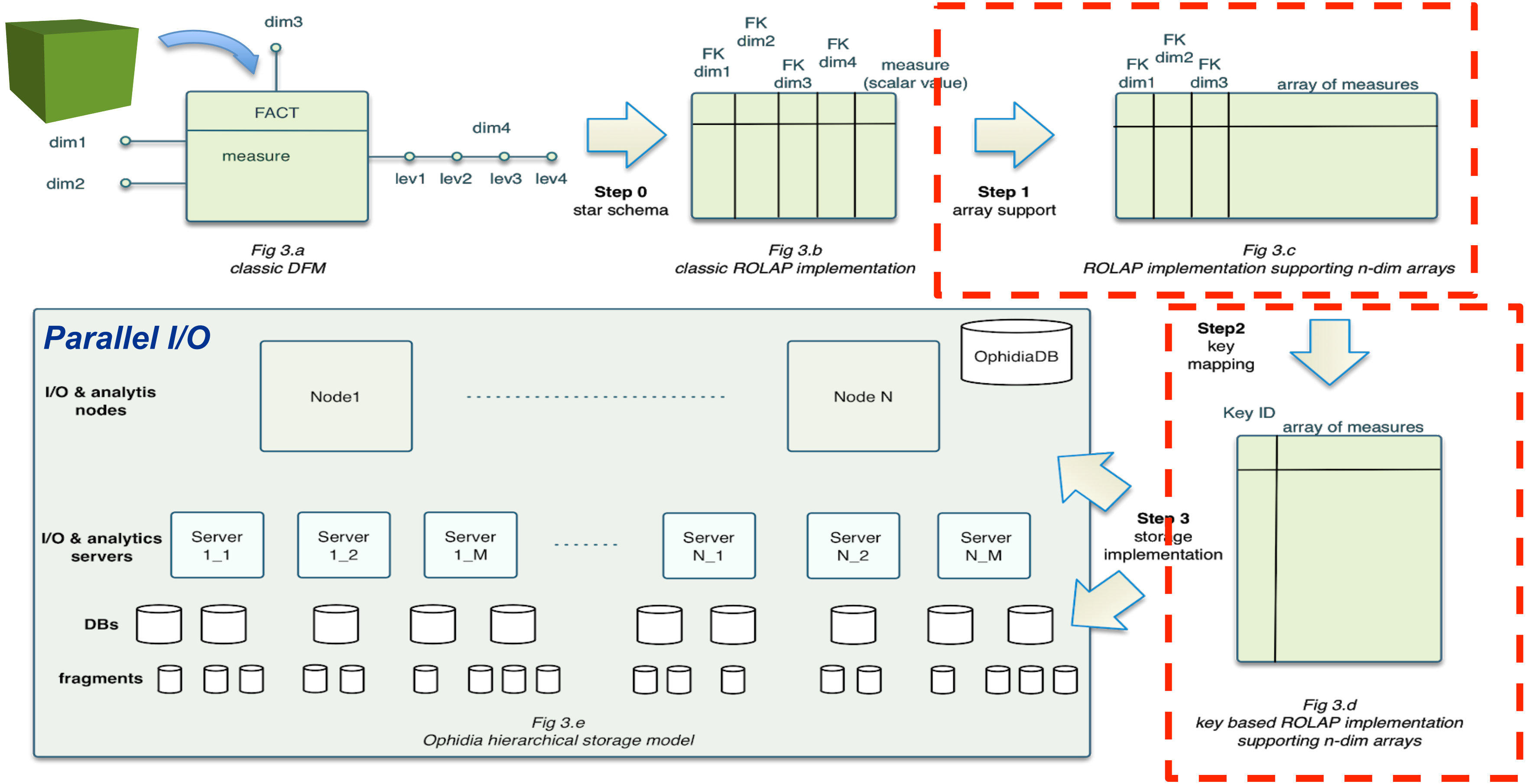


Fig 3.a
classic DFM

Fig 3.b
classic ROLAP implementation

Fig 3.c
ROLAP implementation supporting n-dim arrays

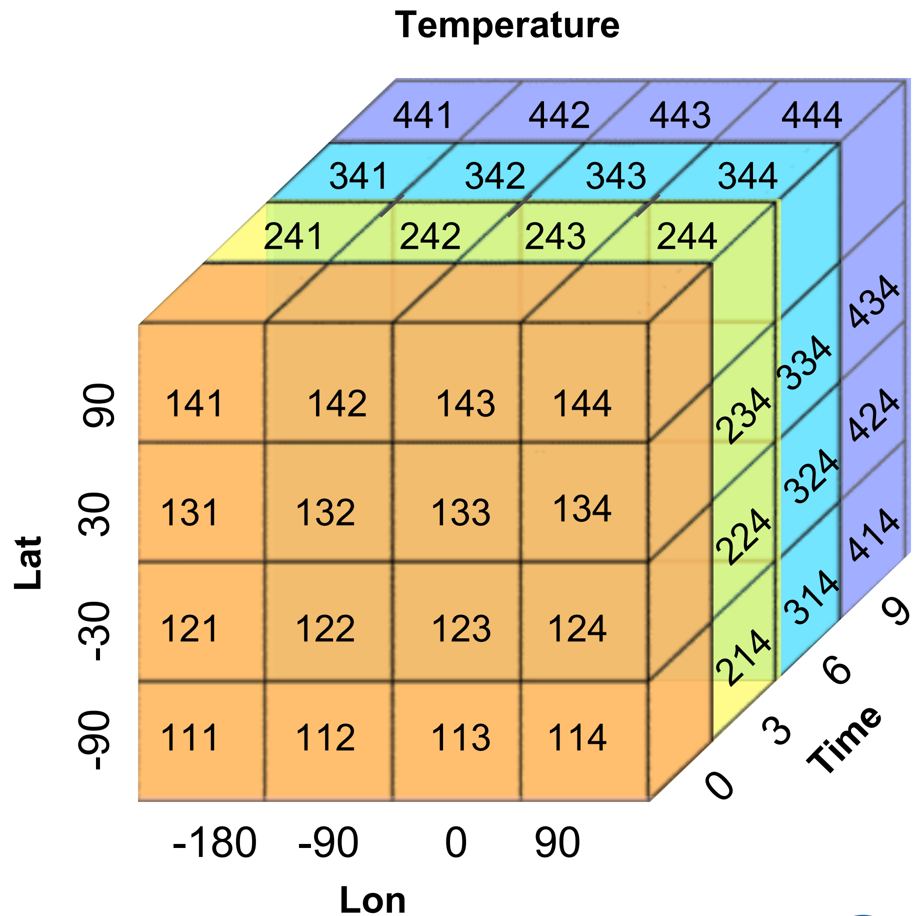
Fig 3.e
Ophidia hierarchical storage model

Fig 3.d
key based ROLAP implementation supporting n-dim arrays

S. Fiore, D. Elia, C. Palazzo, F. Antonio, A. D'Anca, I. Foster and G. Aloisio, "Towards High Performance Data Analytics for Climate Change", ISC High Performance 2019. Lecture Notes in Computer Science, vol. 11887, pp. 240-257, 2019.

From NetCDF to datacube

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
        111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
        211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, “33, 234,
          241, 242, 243, 244,
        ...
```



From NetCDF to Ophidia

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
        111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
        211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, 233, 234,
          241, 242, 243, 244,
        311, 312, 313, 314,
          ...
```

NetCDF

		Temperature			
lat	lon	time[0]	time[1]	time[2]	time[3]
-90	-180	111	211	311	411
-90	-90	112	212	312	412
-90	0	113	213	313	413
-90	90	114	214	314	414
-30	-180	121	221	321	421
-30	-90	122	222	322	422
-30	0	123	223	323	423
-30	90	124	224	324	424
30	-180	131	231	331	431
30	-90	132	232	332	432
30	0	133	233	333	433
30	90	134	234	334	434
90	-180	141	241	341	441
90	-90	142	242	342	442
90	0	143	243	343	443
90	90	144	244	344	444

Ophidia

From NetCDF to Ophidia

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
        111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
        211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, 233, 234,
          241, 242, 243, 244,
        311, 312, 313, 314,
          ...
```

NetCDF

		Temperature			
lat	lon	time[0]	time[1]	time[2]	time[3]
-90	-180	111	211	311	411
-90	-90	112	212	312	412
-90	0	113	213	313	413
-90	90	114	214	314	414
-30	-180	121	221	321	421
-30	-90	122	222	322	422
-30	0	123	223	323	423
-30	90	124	224	324	424
30	-180	131	231	331	431
30	-90	132	232	332	432
30	0	133	233	333	433
30	90	134	234	334	434
90	-180	141	241	341	441
90	-90	142	242	342	442
90	0	143	243	343	443
90	90	144	244	344	444

Ophidia

From NetCDF to Ophidia

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
        111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
        211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, 233, 234,
          241, 242, 243, 244,
        311, 312, 313, 314,
          ...
```

NetCDF

ID	Array			
1	111	211	311	411
2	112	212	312	412
3	113	213	313	413
4	114	214	314	414
5	121	221	321	421
6	122	222	322	422
7	123	223	323	423
8	124	224	324	424
9	131	231	331	431
10	132	232	332	432
11	133	233	333	433
12	134	234	334	434
13	141	241	341	441
14	142	242	342	442
15	143	243	343	443
16	144	244	344	444

Ophidia

From NetCDF to Ophidia

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
       111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
       211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, 233, 234,
          241, 242, 243, 244,
       311, 312, 313, 314,
          ...
```

NetCDF

lat	lon	Temperature			
		time[0]	time[1]	time[2]	time[3]
-90	-180	111	211	311	411
-90	-90	112	212	312	412
-90	0	113	213	313	413
-90	90	114	214	314	414
-30	-180	121	221	321	421
-30	-90	122	222	322	422
-30	0	123	223	323	423
-30	90	124	224	324	424
30	-180	131	231	331	431
30	-90	132	232	332	432
30	0	133	233	333	433
30	90	134	234	334	434
90	-180	141	241	341	441
90	-90	142	242	342	442
90	0	143	243	343	443
90	90	144	244	344	444

Ophidia

From NetCDF to Ophidia

```
netcdf test {
dimensions:
    lat = 4 ;
    lon = 4 ;
    time = UNLIMITED // (4 currently) ;
variables:
    double lon(lon) ;
    double lat(lat) ;
    double time(time) ;
    float Temperature(time, lat, lon) ;
data:
    lon = -180, -90, 0, 90 ;
    lat = -90, -30, 30, 90 ;
    time = 0, 3, 6, 9 ;
    temperature =
        111, 112, 113, 114,
          121, 122, 123, 124,
          131, 132, 133, 134,
          141, 142, 143, 144,
        211, 212, 213, 214,
          221, 222, 223, 224,
          231, 232, 233, 234,
          241, 242, 243, 244,
        311, 312, 313, 314,
          ...
```

NetCDF

FRAG1

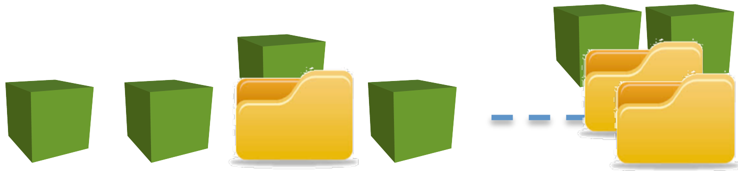
lat	lon	Temperature			
		time[0]	time[1]	time[2]	time[3]
-90	-180	111	211	311	411
-90	-90	112	212	312	412
-90	0	113	213	313	413
-90	90	114	214	314	414
-30	-180	121	221	321	421
-30	-90	122	222	322	422
-30	0	123	223	323	423
-30	90	124	224	324	424

FRAG2

lat	lon	Temperature			
		time[0]	time[1]	time[2]	time[3]
30	-180	131	231	331	431
30	-90	132	232	332	432
30	0	133	233	333	433
30	90	134	234	334	434
90	-180	141	241	341	441
90	-90	142	242	342	442
90	0	143	243	343	443
90	90	144	244	344	444

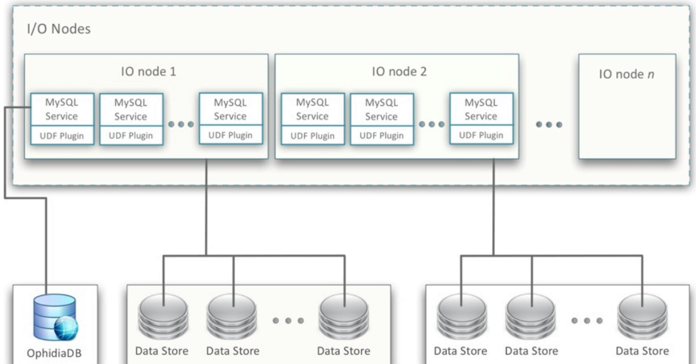
Ophidia

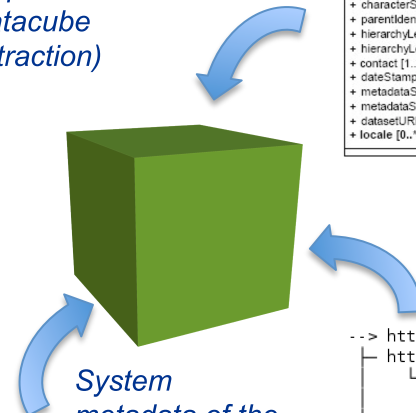
Data abstraction: cube space perspective



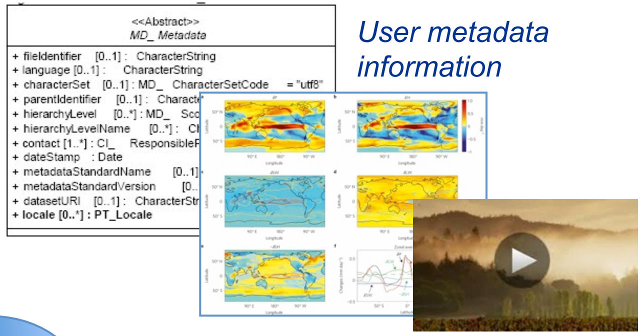
User perspective (datacube abstraction)

System perspective (internal storage representation)





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



User metadata information

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
…	…

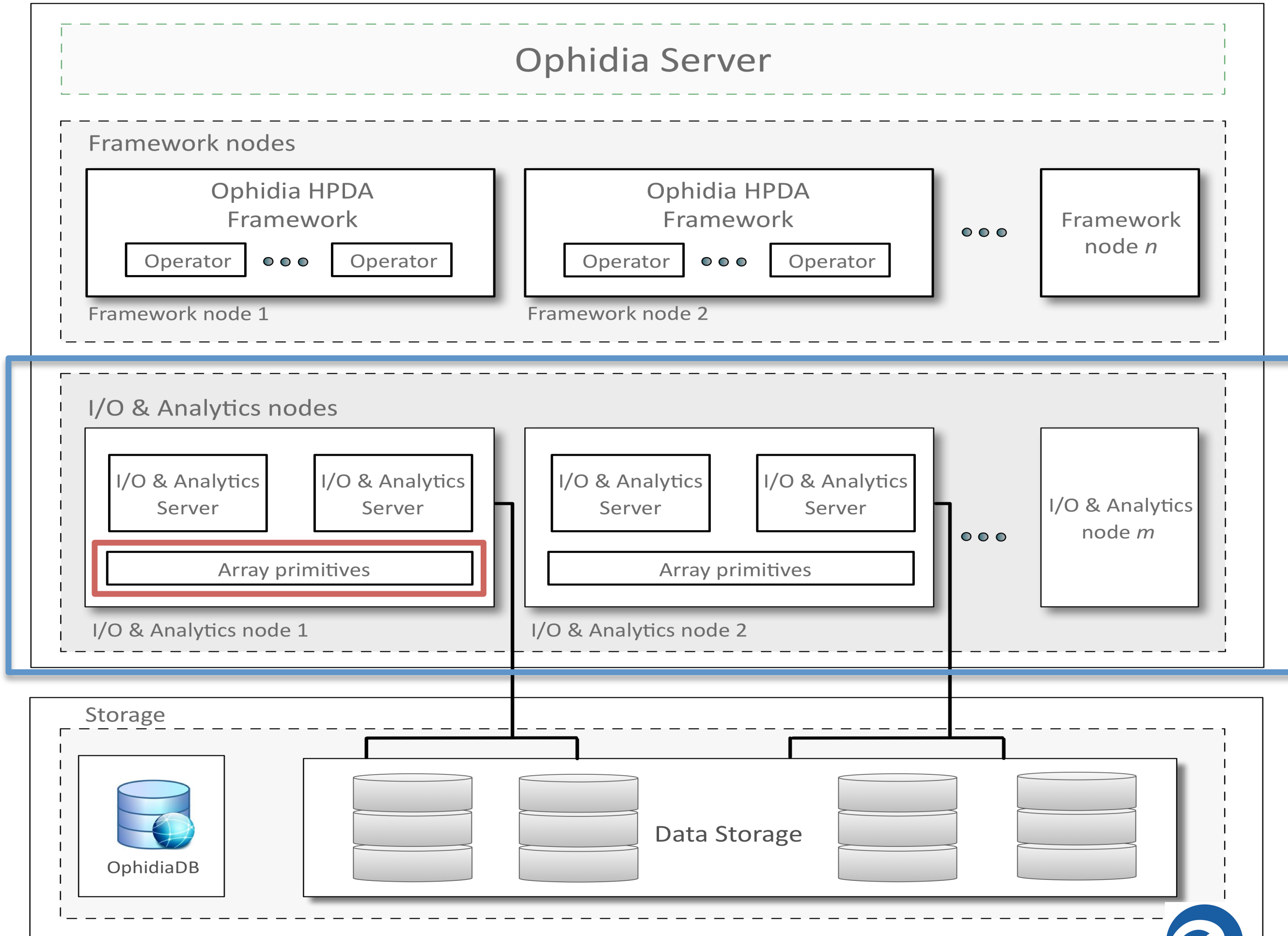
Ophidia architecture: I/O & Analytics layer

*Multiple **I/O & analytics nodes** execute one or more servers*

*Servers run the array-based **primitives** (UDF)*

Servers can transparently interface to different storage back-ends

Support for a native in-memory array-based analytics & I/O engine



Ophidia array-based primitives

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

- data summarisation, sub-setting, predicates evaluation, statistical analysis, array concatenation, algebraic expression, regression, etc.

Primitives come as plugins (UDF) and are applied on a single datacube chunk (fragment)

Primitives can be nested to get more complex functionalities

New primitives can be easily integrated as additional plugins

oph_apply operator to run any primitive on a datacube

```
oph_apply(oph_predicate(measure,'x-298.15','>0','1','0'))
```

Ophidia Primitives documentation: http://ophidia.cmcc.it/documentation/users/primitives/index.html

Array based primitives: nesting feature

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

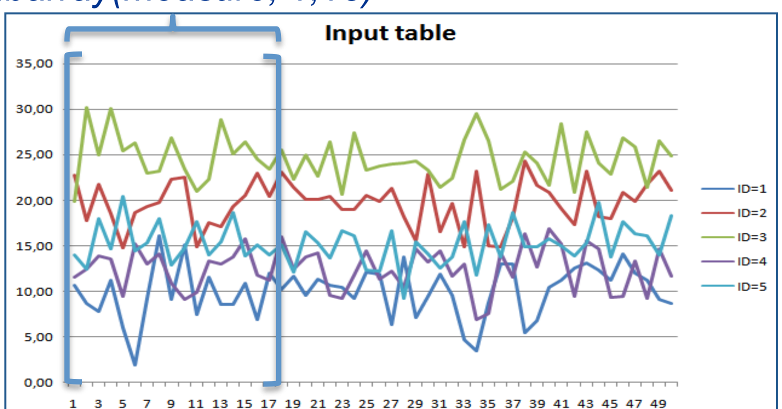
Single chunk or fragment (input)

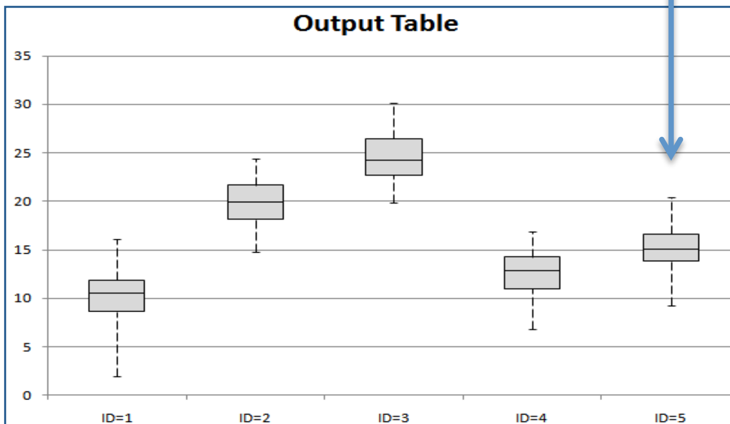
INPUT TABLE 5 tuples x 50 elements										
ID	**MEASURE**									
1	10,73	8,66	7,83	11,20	6,02	1,95	...	16,11	...	8,70
2	22,85	17,84	21,82	18,57	14,81	18,71	...	19,83	...	21,13
3	19,89	30,17	24,95	30,07	25,40	26,31	...	23,18	...	24,82
4	11,60	12,49	13,91	13,53	9,48	15,27	...	14,17	...	11,66
5	13,94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

Single chunk or fragment (output)

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

subarray(measure, 1,18)





Array based primitives: oph_aggregate

oph_aggregate*(measure,"oph_avg")*

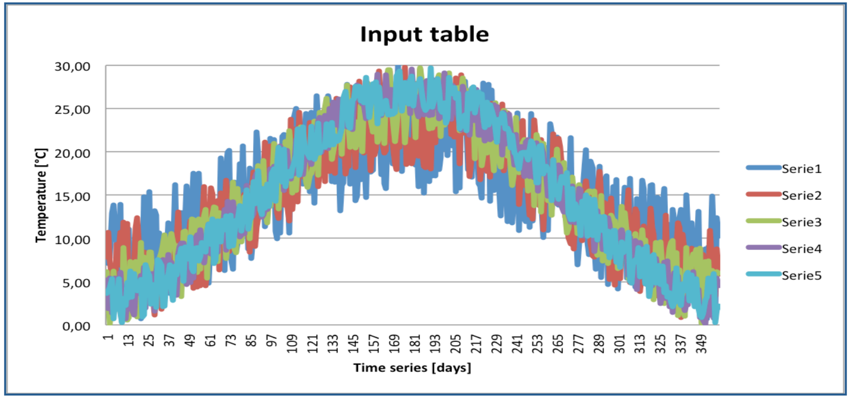
INPUT TABLE 5 tuples x 360 elements										
ID	**MEASURE**									
1	8,40	7,73	7,36	12,68	13,34	11,17	9,09	2,04	...	7,75
2	7,85	10,71	7,23	5,14	4,68	2,61	9,17	8,50	...	6,57
3	6,40	3,48	0,44	2,81	6,16	2,01	3,61	3,83	...	5,88
4	5,60	4,68	5,54	5,84	5,47	5,37	5,30	7,24	...	3,06
5	3,55	4,10	4,59	5,07	6,97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

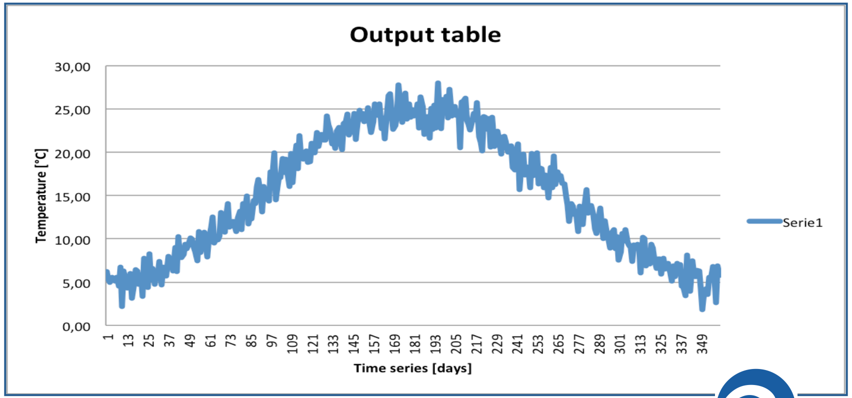
Single chunk or fragment (input)

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)



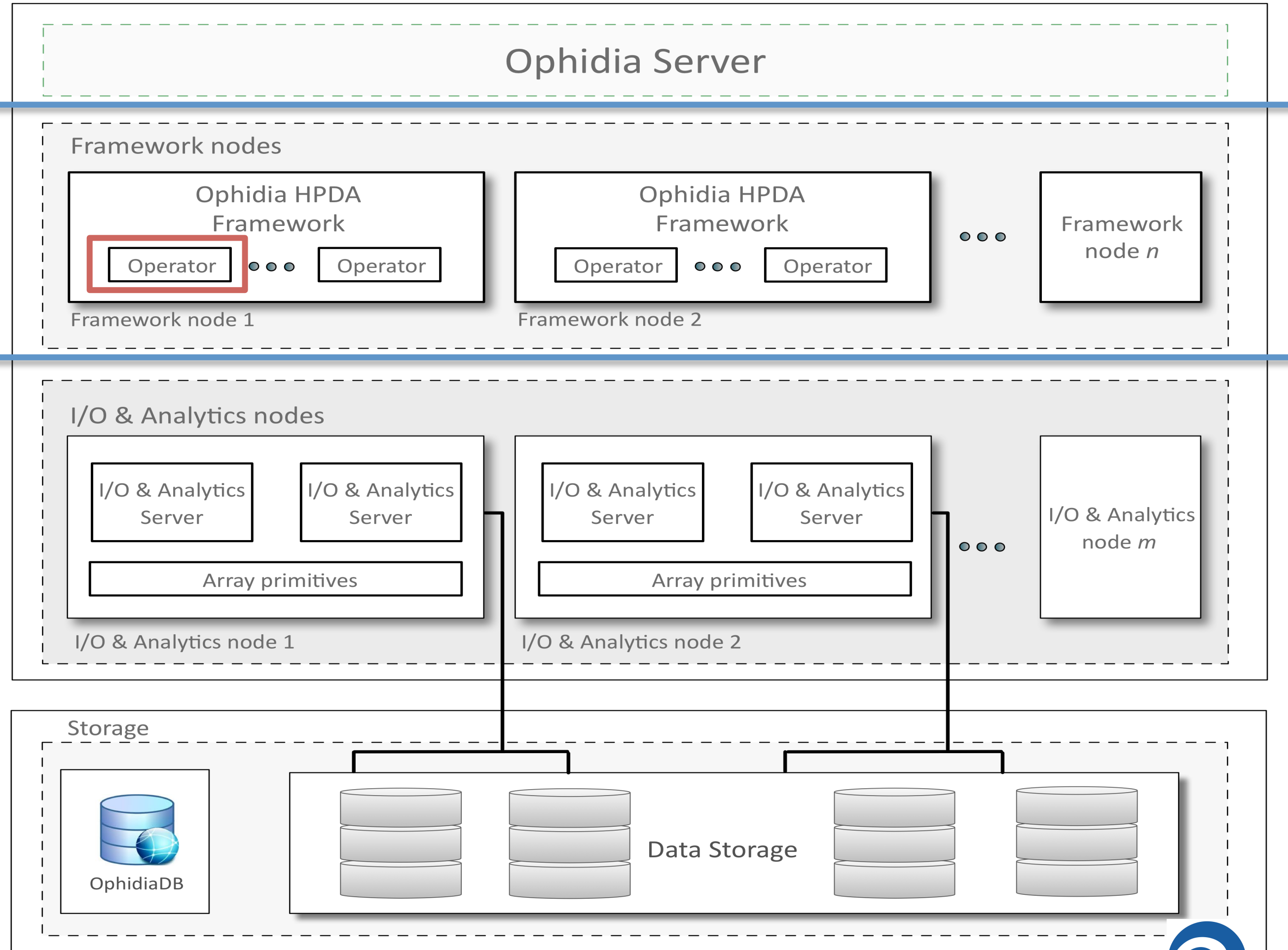


Ophidia architecture: framework layer

The Ophidia analytics framework can be executed with multiple processes/threads

*Provides the environment for the execution of parallel MPI/Pthread-based **operators***

*Operators manipulate the entire set of fragments associated to a **whole datacube***



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 processing

Ophidia operators documentation: http://ophidia.cmcc.it/documentation/users/operators/index.html

Ophidia “data” operators

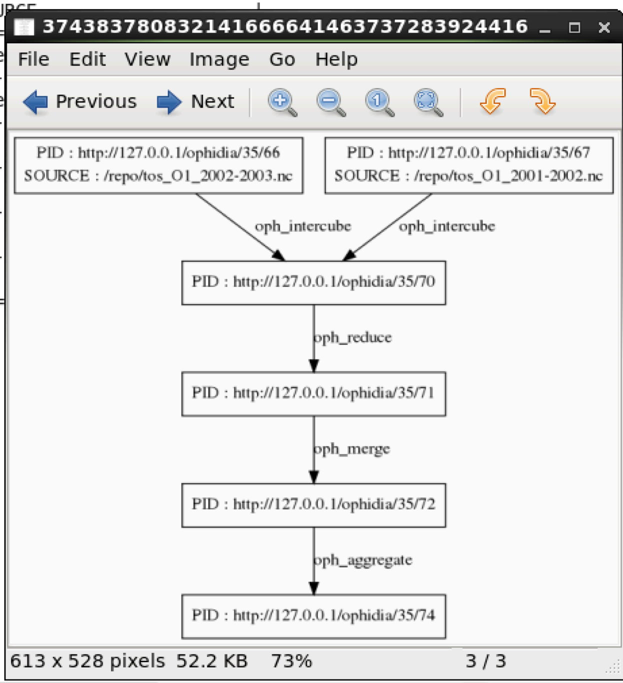
```
[37..4416] >> oph_explorecube cube=http://127.0.0.1/ophidia/35/67;subset_dims=lat|lon|time;subset_filter=39:42|15:19|1:275;show_time=yes;
[Request]:
operator=oph_explorecube;cube=http://127.0.0.1/ophidia/35/67;subset_dims=lat|lon|time;subset_filter=39:42|15:19|1:275;show_time=yes;sessionid=http://127.0.0.1/ophidia/sessions/3
7438378083214166664146373728392441 6/experiment;exec_mode=sync;ncores=1;cwd=/;

[JobID]:
http://127.0.0.1/ophidia/sessions/37438378083214166664146373728392441 6/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
```

Ophidia “metadata” operators

```
[37..4416] >> oph_cubeio
[Request]:
operator=oph_cubeio;sessionid=http://127.0.0.1/ophidia/sessions/374383780832141666641463737283924416/experiment;exec_mode=sync;ncores=1;cube=http://127.0.0.1/ophidia/35/74;cwd=/
;

[JobID]:
http://127.0.0.1/ophidia/sessions/374383780832141666641463737283924416/experiment?82#176

[Response]:
Cube Provenance
---------------

+===================================================================+===============+=================================+=========================+
| INPUT CUBE                                                        | OPERATION     | OUTPUT CUBE                     | SOURCE
+===================================================================+===============+=================================+===
|                                                                   | ROOT          | http://127.0.0.1/ophidia/35/66  | /re
|-------------------------------------------------------------------|---------------|---------------------------------|---
|                                                                   | ROOT          | http://127.0.0.1/ophidia/35/67  | /re
|-------------------------------------------------------------------|---------------|---------------------------------|---
| http://127.0.0.1/ophidia/35/66 - http://127.0.0.1/ophidia/35/67   | oph_intercube | http://127.0.0.1/ophidia/35/70  |
|-------------------------------------------------------------------|---------------|---------------------------------|---
| http://127.0.0.1/ophidia/35/70                                    | oph_reduce    | http://127.0.0.1/ophidia/35/71  |
|-------------------------------------------------------------------|---------------|---------------------------------|---
| http://127.0.0.1/ophidia/35/71                                    | oph_merge     | http://127.0.0.1/ophidia/35/72  |
|-------------------------------------------------------------------|---------------|---------------------------------|---
| http://127.0.0.1/ophidia/35/72                                    | oph_aggregate | http://127.0.0.1/ophidia/35/74  |
+===================================================================+===============+=================================+===

Cube Provenance Graph
---------------------

Directed Graph DOT string :
digraph DG {
        node    [shape=box]

        0       [label="PID : http://127.0.0.1/ophidia/35/74\n"]
        1       [label="PID : http://127.0.0.1/ophidia/35/72\n"]
        2       [label="PID : http://127.0.0.1/ophidia/35/71\n"]
        3       [label="PID : http://127.0.0.1/ophidia/35/70\n"]
        4       [label="PID : http://127.0.0.1/ophidia/35/66\nSOURCE : /repo/tos_O1_2002-2003.nc\n"]
        5       [label="PID : http://127.0.0.1/ophidia/35/67\nSOURCE : /repo/tos_O1_2001-2002.nc\n"]

        1->0    [label="oph_aggregate"]
        2->1    [label="oph_merge"]
```



Ophidia architecture: front-end layer

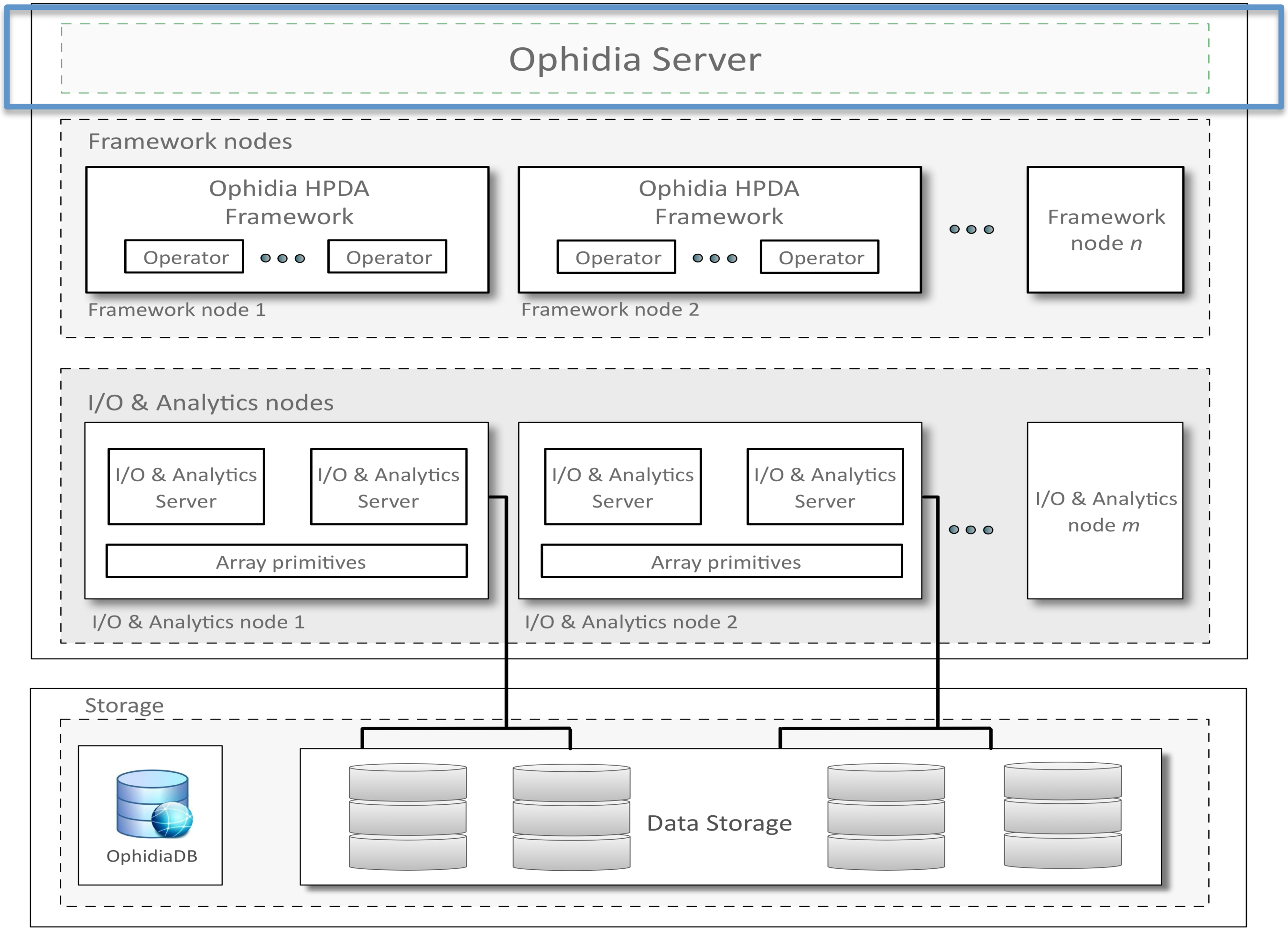
Multi-interface *server front-end*

*Manages user **authN/ authZ, sessions** and requests*

*Manages **task/ workflow** execution*

Remote interactions with:

- *oph_term (CLI)*
- *WPS clients*
- *Python modules*



Three levels of parallelism

Datacube-level parallelism

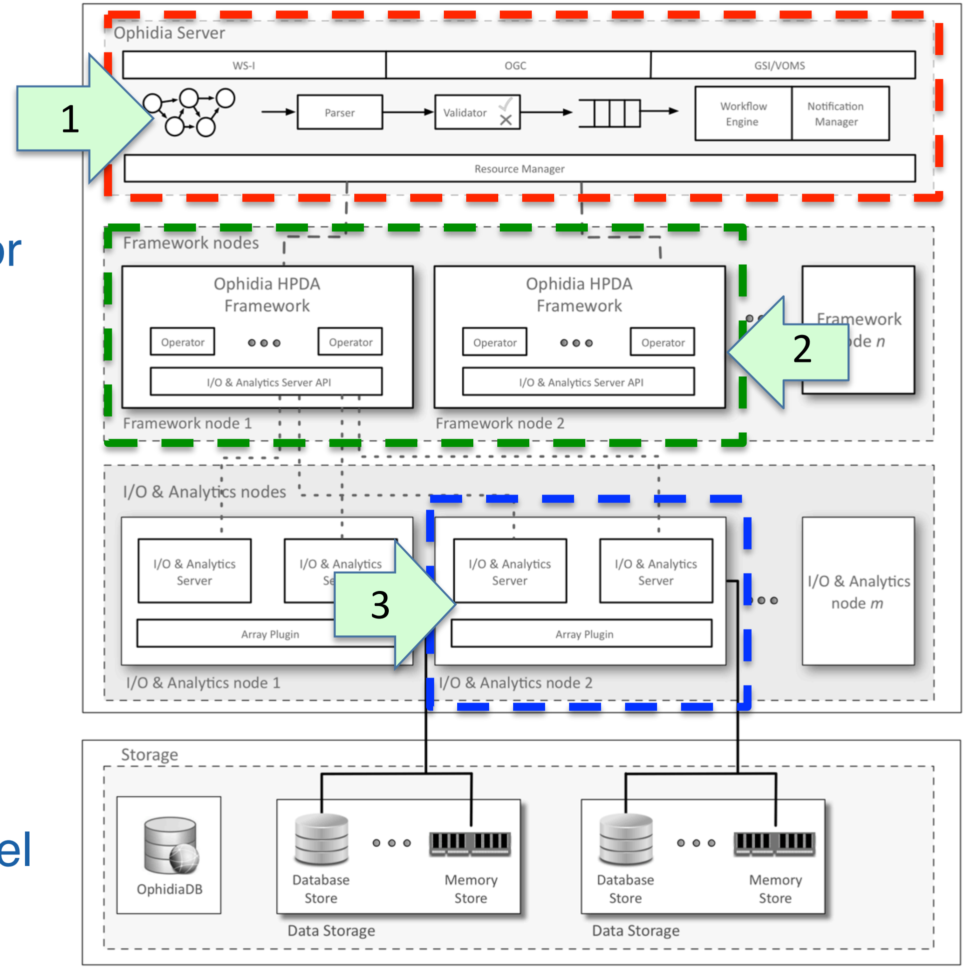
- HTC paradigm
- At the front-end level
- Based on the "massive" operator concept

Framework-level parallelism

- HPC paradigm
- MPI/Pthread
- At the HPDA framework level

Fragment-level parallelism

- OpenMP based
- At the I/O & analytics server level



On-demand instantiation of an Ophidia cluster

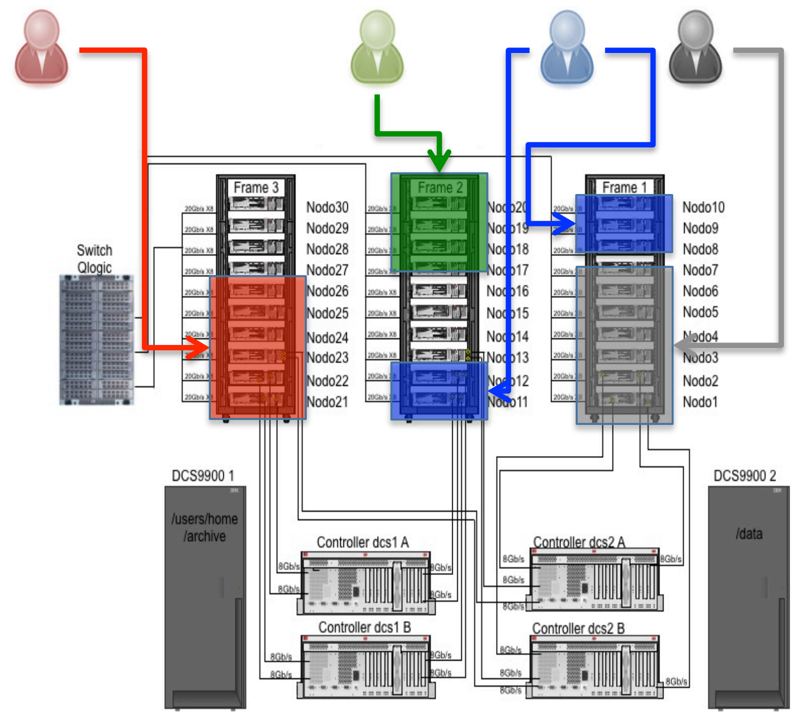
Target environment: HPC cluster

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



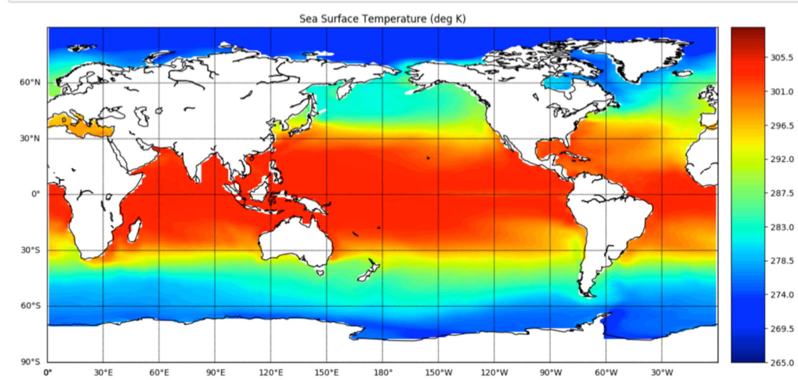
Python programmatic access to Ophidia

PyOphidia is a GPLv3-licensed Python module to interact with the Ophidia framework.

It provides a programmatic access to Ophidia features, allowing:

- Submission of commands to the Ophidia Server and retrieval of the results
- Management of (remote) data objects in the form of datacubes
- Easy exploitation from Jupyter Notebooks and integration with other Python modules

```
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')
```



Export result to NetCDF file

```
]: mycube3.exportnc2(output_path='/home/' + cube.Cube.client.username,export_metadata='yes')
```

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')
```

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
            )

    ]: 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

Summary

- ✔ *Scientific data management and analytics pose challenges requiring novel and efficient software solution*
- ✔ *Joining HPC and data-intensive analytics is an enabling factor for scientific applications*
- ✔ *The **Ophidia HPDA framework** addresses challenges for scientific analysis, through:*
  - ***Scalable** architecture*
  - *Data **distribution** and partitioning*
  - ***Parallel** (MPI/Pthread-based) **operators***
  - ***HPC-oriented deployment***
- ✔ *PyOphidia module hide the complexity of HPC infrastructure, provides a user-friendly interface and can be easily exploited in Jupyter Notebooks*

References and further readings

- *D. A. Reed and J. Dongarra. (2015). Exascale computing and big data. Commun. ACM 58, 7 (July 2015), 56–68.*
- *Jha, S., Qiu, J., Luckow, A., Mantha, P., & Fox, G. C. (2014). A tale of two data-intensive paradigms: Applications, abstractions, and architectures. In 2014 IEEE Int. Congress on Big Data, 645-652.*
- *Asch, M., et al. (2018). Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. Int. J. High Perform. Comput. Appl., 32(4), 435-479.*
- *Luca Cinquini, et al. (2014). The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. Future Gener. Comput. Syst. 36: 400-417.*
- *GMD topical editors (Eds.), V. Eyring (coordinator) (2012). Coupled Model Intercomparison Project Phase 6 (CMIP6) Experimental Design and Organization [Special Issue]. Geosci. Model Dev. https://gmd.copernicus.org/articles/special_issue590.html*
- *G. Aloisio, S. Fiore, I. Foster, D. Williams (2013). 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).*
- *S. Fiore, A. D'Anca, C. Palazzo, I. T. Foster, D. N. Williams, G. Aloisio (2013). Ophidia: Toward Big Data Analytics for eScience. ICCS 2013, volume 18 of Procedia Computer Science, pp. 2376-2385.*
- *S. Fiore, A. D'Anca, D. Elia, C. Palazzo, I. Foster, D. Williams, G. Aloisio (2014). "Ophidia: A Full Software Stack for Scientific Data Analytics", proc. of the 2014 Int. Conference on High Performance Computing & Simulation (HPCS 2014), pp. 343-350.*
- *S. Fiore, D. Elia, C. Palazzo, F. Antonio, A. D'Anca, I. Foster and G. Aloisio (2019), "Towards High Performance Data Analytics for Climate Change", ISC High Performance 2019. Lecture Notes in Computer Science, vol. 11887, pp. 240-257.*
- *D. Elia, S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. N. Williams, G. Aloisio (2016). "An in-memory based framework for scientific data analytics". In Proc. of the ACM Int. Conference on Computing Frontiers (CF '16), pp. 424-429.*

Thank you!

This training has been organised in the context of the ESiWACE2 project:

ESiWACE2 has received funding from 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

Ophidia website: *http://ophidia.cmcc.it*

Contact: *ophidia-info AT cmcc.it*

cmcc

Centro Euro-Mediterraneo sui Cambiamenti Climatici