Ophidia: a big data analytics framework for climate change

Eng. Alessandro D’Anca
Dr. Sandro Fiore and Prof. Giovanni Aloisio
CMCC Foundation
Advanced Scientific Computing Division

on behalf of the Ophidia Team
Outline

• Analytics requirements and big data challenges in the climate context

• Ophidia
  – Architecture v1.0
    • Storage model
    • Primitives
    • Data and metadata operators
  – Architecture v2.0
    • Workflow support
      – Some real use cases

• Useful link & documentation
  – Website, github, youtube, pypi, ...
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)
- Maps generation
- Ensemble analysis
- Data analytics workflow support

But also…
- Performance
- re-usability
- extensibility
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.

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

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

Exploits a multidimensional data model providing the data cube abstraction for access and analysis of scientific n-dimensional data.
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
**Ophidia Architecture: end-user view**

**Oph_Term**: a commands interpreter with no GUI (like bash), serving as a client for the Ophidia framework

**Ophidia framework**: Server-side processing

Through the *oph_term* we “send” commands to the framework
Ophidia Architecture (sw stack view)

- **Front end**
  - Compute layer
  - I/O layer
  - I/O server instance
  - Storage layer
  - System catalog

- **Compute layer**
  - Compute nodes
    - Compute node 1
    - Compute node 2
    - ... Compute node n

- **I/O layer**
  - I/O Nodes
    - I/O node 1
    - I/O node 2
    - ... I/O node n

- **Storage layer**
  - Storage
    - Data Stores
    - OphidiaDB

- **System catalog**
  - OphidiaDB

- **Analytics Framework**
  - Array-based primitives
  - New storage model
  - Partitioning/hierarchical data mng

- **Standard interfaces**
  - Declarative language

- **Ophidia Server**
Storage model and chunks distribution

Front end

Compute layer

I/O layer

I/O server instance

Storage layer

System catalog

OPHIDIA Server

Compute node 1

Compute node 2

Compute node n

I/O Nodes

I/O node 1

I/O node 2

I/O node n

MySQL Service

MySQL Service

MySQL Service

MySQL Service

MySQL Service

MySQL Service

MySQL Service

MySQL Service

UDF Plugin

UDF Plugin

UDF Plugin

UDF Plugin

UDF Plugin

UDF Plugin

UDF Plugin

UDF Plugin

Storage

Data Store

Data Store

Data Store

Data Store

Data Store

Data Store

Data Store

New storage model

Partitioning/hierarchical data mng

OphidiaDB
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

Parallel I/O

I/O nodes
I/O servers
I/O node1
I/O node N

I/O server 1_1
I/O server 1_2
I/O server 1_M
I/O server N_1
I/O server N_2
I/O server N_M

DBs

fragments

Fig 1.a classic DFM
Fig 1.b classic ROLAP implementation
Fig 1.c ROLAP implementation supporting n-dim arrays
Fig 1.e Ophidia hierarchical storage model

Fig 1.d key based ROLAP implementation supporting n-dim arrays
Array-based primitives

Front end

Compute layer

I/O layer

I/O server instance

Storage layer

System catalog
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)

- **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")

Single chunk or fragment (input)

Single chunk or fragment (output)
Array-based primitives: OPH_MATH support

**OPH_MATH_FUNCTION MACROS**

<table>
<thead>
<tr>
<th>OPH_MATH_ABS</th>
<th>OPH_MATH_DEGREES</th>
<th>OPH_MATH_RAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH_MATH_ACOS</td>
<td>OPH_MATH_EXP</td>
<td>OPH_MATH_ROUND</td>
</tr>
<tr>
<td>OPH_MATH_ASIN</td>
<td>OPH_MATH_FLOOR</td>
<td><strong>OPH_MATH_SIGN</strong></td>
</tr>
<tr>
<td>OPH_MATH_ATAN</td>
<td>OPH_MATH_LN</td>
<td>OPH_MATH_SIN</td>
</tr>
<tr>
<td>OPH_MATH_CEIL</td>
<td>OPH_MATH_LOG10</td>
<td>OPH_MATH_SQRT</td>
</tr>
<tr>
<td>OPH_MATH_COS</td>
<td>OPH_MATH_LOG2</td>
<td>OPH_MATH_TAN</td>
</tr>
<tr>
<td>OPH_MATH_COT</td>
<td>OPH_MATH_RADIANS</td>
<td>...</td>
</tr>
</tbody>
</table>
Array based primitives: OPH_BOXPLOT

\texttt{oph_boxplot}(\texttt{measure, "OPH\_DOUBLE"})

**Single chunk or fragment (input)**

**INPUTTABLE 5 tuples x 50 elements**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.73</td>
</tr>
<tr>
<td>2</td>
<td>22.85</td>
</tr>
<tr>
<td>3</td>
<td>19.89</td>
</tr>
<tr>
<td>4</td>
<td>11.60</td>
</tr>
<tr>
<td>5</td>
<td>13.94</td>
</tr>
</tbody>
</table>

**Single chunk or fragment (output)**

**OUTPUTTABLE 5 tuples x 5 elements (summary)**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.05</td>
</tr>
<tr>
<td>3</td>
<td>19.89</td>
</tr>
<tr>
<td>4</td>
<td>6.87</td>
</tr>
<tr>
<td>5</td>
<td>9.25</td>
</tr>
</tbody>
</table>
Array based primitives: nesting feature

\[ oph\_boxplot(oph\_subarray(oph\_uncompress(measure), 1,18), \text{"OPH\_DOUBLE"}) \]

**Single chunk or fragment (input)**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.73 8.66 7.83 11.20 6.02 1.95 ... 16.11 ... 8.70</td>
</tr>
<tr>
<td>2</td>
<td>22.85 17.84 21.82 18.57 14.81 18.71 ... 19.83 ... 21.13</td>
</tr>
<tr>
<td>3</td>
<td>19.89 30.17 24.95 30.07 25.40 26.31 ... 23.18 ... 24.82</td>
</tr>
<tr>
<td>4</td>
<td>11.60 12.49 13.91 13.53 9.48 15.27 ... 14.17 ... 11.66</td>
</tr>
<tr>
<td>5</td>
<td>13.94 12.45 17.95 14.70 20.41 14.40 ... 18.00 ... 18.30</td>
</tr>
</tbody>
</table>

**Single chunk or fragment (output)**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95 8.64 10.47 11.87 16.11</td>
</tr>
<tr>
<td>3</td>
<td>19.89 22.74 24.22 26.45 30.17</td>
</tr>
<tr>
<td>4</td>
<td>6.87 10.99 12.85 14.28 16.93</td>
</tr>
<tr>
<td>5</td>
<td>9.23 15.87 15.05 16.61 20.41</td>
</tr>
</tbody>
</table>
Array based primitives: oph_aggregate

**oph_aggregate**(measure,"oph_avg")

**Single chunk or fragment (input)**

**INPUT TABLE 5 tuples x 360 elements**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8,40</td>
</tr>
<tr>
<td></td>
<td>7,73</td>
</tr>
<tr>
<td></td>
<td>7,36</td>
</tr>
<tr>
<td></td>
<td>12,68</td>
</tr>
<tr>
<td></td>
<td>13,34</td>
</tr>
<tr>
<td></td>
<td>11,17</td>
</tr>
<tr>
<td></td>
<td>9,09</td>
</tr>
<tr>
<td></td>
<td>2,04</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>7,85</td>
</tr>
<tr>
<td></td>
<td>10,71</td>
</tr>
<tr>
<td></td>
<td>7,23</td>
</tr>
<tr>
<td></td>
<td>5,14</td>
</tr>
<tr>
<td></td>
<td>4,68</td>
</tr>
<tr>
<td></td>
<td>2,61</td>
</tr>
<tr>
<td></td>
<td>9,17</td>
</tr>
<tr>
<td></td>
<td>8,50</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>6,40</td>
</tr>
<tr>
<td></td>
<td>3,48</td>
</tr>
<tr>
<td></td>
<td>0,44</td>
</tr>
<tr>
<td></td>
<td>2,81</td>
</tr>
<tr>
<td></td>
<td>6,16</td>
</tr>
<tr>
<td></td>
<td>2,01</td>
</tr>
<tr>
<td></td>
<td>3,61</td>
</tr>
<tr>
<td></td>
<td>3,83</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>5,60</td>
</tr>
<tr>
<td></td>
<td>4,68</td>
</tr>
<tr>
<td></td>
<td>5,54</td>
</tr>
<tr>
<td></td>
<td>5,84</td>
</tr>
<tr>
<td></td>
<td>5,47</td>
</tr>
<tr>
<td></td>
<td>5,37</td>
</tr>
<tr>
<td></td>
<td>5,30</td>
</tr>
<tr>
<td></td>
<td>7,24</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>3,55</td>
</tr>
<tr>
<td></td>
<td>4,10</td>
</tr>
<tr>
<td></td>
<td>4,59</td>
</tr>
<tr>
<td></td>
<td>5,07</td>
</tr>
<tr>
<td></td>
<td>6,97</td>
</tr>
<tr>
<td></td>
<td>2,07</td>
</tr>
<tr>
<td></td>
<td>3,06</td>
</tr>
<tr>
<td></td>
<td>3,06</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

**Vertical aggregation**

**OUTPUT TABLE 1 tuple x 360 elements**

<table>
<thead>
<tr>
<th>ID</th>
<th>MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6,25</td>
</tr>
<tr>
<td></td>
<td>5,35</td>
</tr>
<tr>
<td></td>
<td>5,00</td>
</tr>
<tr>
<td></td>
<td>5,57</td>
</tr>
<tr>
<td></td>
<td>5,41</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>5,11</td>
</tr>
</tbody>
</table>

**Single chunk or fragment (output)**

**Input table**

**Output table**
Analytics framework and operators

Front end

Compute layer

I/O layer

I/O server instance

Storage layer

System catalog

Analytics Framework

OPHIDIA Server

Compute nodes

I/O Nodes

Storage

OphidiaDB

MySQL Service UDF Plugin

MySQL Service UDF Plugin

MySQL Service UDF Plugin

MySQL Service UDF Plugin

MySQL Service UDF Plugin

MySQL Service UDF Plugin

I/O node 1

I/O node 2

I/O node n

Compute node 1

Compute node 2

Compute node n

Data Store

Data Store

Data Store

Data Store

Data Store

Data Store
## Datacube abstraction and operators (about 50)

### Data processing

<table>
<thead>
<tr>
<th>OPERATOR NAME</th>
<th>OPERATOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH_APPLY(datacube_in,</td>
<td>Creates the datacube_out by applying the array-based primitive to the datacube_in</td>
</tr>
<tr>
<td>datacube_out, array_based_primitive)</td>
<td></td>
</tr>
<tr>
<td>OPH_DUPLICATE(datacube_in,</td>
<td>Creates a copy of the datacube_in in the datacube_out</td>
</tr>
<tr>
<td>datacube_out)</td>
<td></td>
</tr>
<tr>
<td>OPH_SUBSET(datacube_in, subset_string, datacube_out)</td>
<td>Creates the datacube_out by doing a sub-setting of the datacube_in by applying the subset_string</td>
</tr>
<tr>
<td>OPH_MERGE(datacube_in, merge_param, datacube_out)</td>
<td>Creates the datacube_out by merging groups of merge_param fragments from datacube_in</td>
</tr>
<tr>
<td>OPH_SPLIT(datacube_in, split_param, datacube_out)</td>
<td>Creates the datacube_out by splitting into groups of split_param fragments each fragment of the datacube_in</td>
</tr>
<tr>
<td>OPH_INTERCOMPARISON(datacube_in1, datacube_in2, datacube_out)</td>
<td>Creates the datacube_out which is the element-wise difference between datacube_in1 and datacube_in2</td>
</tr>
<tr>
<td>OPH_DELETE(datacube_in)</td>
<td>Removes the datacube_in</td>
</tr>
</tbody>
</table>

### Import/Export

<table>
<thead>
<tr>
<th>OPERATOR NAME</th>
<th>OPERATOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH_EXPORT_NC(datacube_in, file_out)</td>
<td>Exports the datacube_in data into the file_out NetCDF file.</td>
</tr>
<tr>
<td>OPH_IMPORT_NC(file_in, datacube_out)</td>
<td>Imports the data stored into the file_in NetCDF file into the new datacube_in datacube</td>
</tr>
</tbody>
</table>

### Data Access

<table>
<thead>
<tr>
<th>OPERATOR NAME</th>
<th>OPERATOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH_INSPECT_FRAG(datacube_in, fragment_in)</td>
<td>Inspects the data stored in the fragment_in from the datacube_in</td>
</tr>
<tr>
<td>OPH_PUBLISH(datacube_in)</td>
<td>Publishes the datacube_in fragments into HTML pages</td>
</tr>
</tbody>
</table>

### Metadata management

<table>
<thead>
<tr>
<th>OPERATOR NAME</th>
<th>OPERATOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH_CUBE_ELEMENTS(datacube_in)</td>
<td>Provides the total number of the elements in the datacube_in</td>
</tr>
<tr>
<td>OPH_CUBE_SIZE(datacube_in)</td>
<td>Provides the disk space occupied by the datacube_in</td>
</tr>
<tr>
<td>OPH_LIST(void)</td>
<td>Provides the list of available datacubes.</td>
</tr>
<tr>
<td>OPH_CUBEINFO(datacube_in)</td>
<td>Provides the provenance information related to the datacube_in</td>
</tr>
<tr>
<td>OPH_FIND(search_param)</td>
<td>Provides the list of datacubes matching the search_param criteria</td>
</tr>
</tbody>
</table>
The analytics framework: “data” operators

### Input Data Cube

<table>
<thead>
<tr>
<th>Fragment1 – 10 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95</td>
<td>8.64</td>
</tr>
<tr>
<td>2</td>
<td>14.81</td>
<td>18.14</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>6.87</td>
<td>10.99</td>
</tr>
</tbody>
</table>

### Output Data Cube

<table>
<thead>
<tr>
<th>Fragment1 – 10 Tuple x 1</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>24.35</td>
<td></td>
</tr>
</tbody>
</table>

### Output (Input) Data Cube

<table>
<thead>
<tr>
<th>Fragment1 – 1 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95</td>
<td>8.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment2 – 1 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14.81</td>
<td>18.14</td>
</tr>
</tbody>
</table>

### Input (Output) Data Cube

<table>
<thead>
<tr>
<th>Fragment1 – 10 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.87</td>
<td>10.99</td>
</tr>
</tbody>
</table>

### Output Data Cube

<table>
<thead>
<tr>
<th>Fragment10 – 2 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95</td>
<td>8.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment10 – 2 Tuple x 10 Elements</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14.81</td>
<td>18.14</td>
</tr>
</tbody>
</table>

- **Reduce All Max**: Reduces each tuple to a single value by applying a specified operation (e.g., maximum).
- **Aggregate All Max**: Aggregates the data by combining tuples with the same ID into a single tuple.
- **Subset by Filter**: Filters the input data based on a given criterion (e.g., selecting only those tuples that meet a specific condition).
- **Harvest by Fragment**: Groups data into fragments, which can be processed independently or in parallel.
Massive data reduction workflow

<table>
<thead>
<tr>
<th>Fragment</th>
<th>ID</th>
<th>Measure</th>
<th>ID</th>
<th>Measure</th>
<th>ID</th>
<th>Measure</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1,95</td>
<td>1</td>
<td>8,64</td>
<td>1</td>
<td>16,11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>14,81</td>
<td>2</td>
<td>18,14</td>
<td>2</td>
<td>24,35</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6,87</td>
<td></td>
<td>10,99</td>
<td>1</td>
<td>16,93</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IDEAL PATH

"3"D base cuboid

"0"D apex cuboid

REDUCE ALL MAX

Merging and reducing the fragments:

<table>
<thead>
<tr>
<th>Fragment</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>24,35</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>19,78</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>24,35</td>
</tr>
</tbody>
</table>

Real path through 4 Ophidia operators

Massive data reduction workflow

<table>
<thead>
<tr>
<th>Fragment</th>
<th>ID</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>24,35</td>
</tr>
</tbody>
</table>
The analytics framework: “metadata” operators

[37..4416] >> oph_cubeschema cube=http://127.0.0.1/ophidia/35/67;
[Request]:
operator=oph_cubeschema;cube=http://127.0.0.1/ophidia/35/67;sessionid=http://127.0.0.1/ophidia/sessions/37438378083214166641463737283924416/experiment;exec_mode=sync;ncores=1;
wd=/;

[JobID]:
http://127.0.0.1/ophidia/sessions/37438378083214166641463737283924416/experiment?89#190

[Response]:
Datacube Information

<table>
<thead>
<tr>
<th>PID</th>
<th>CREATION DATE</th>
<th>MEASURE</th>
<th>MEASURE TYPE</th>
<th>LEVEL</th>
<th>NUMBER OF FRAGMENTS</th>
<th>SOURCE FILE</th>
</tr>
</thead>
</table>

Datacube Additional Information

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>HOST x CU</th>
<th>DBMS x HD</th>
<th>DATABASES x DB</th>
<th>FRAGMENTS x DATAB</th>
<th>ROWS x FRAGME</th>
<th>ELEMENTS x R</th>
<th>COMPRESS</th>
<th>CUBE SI</th>
<th>UNI</th>
<th>NUMBER OF ELEME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sea Surface Temperature 2001-2002</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>170</td>
<td>180</td>
<td>24</td>
<td>no</td>
<td>3.43444</td>
<td>MB</td>
<td>734400</td>
</tr>
</tbody>
</table>

Dimension Information

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>SIZE</th>
<th>HIERARCHY</th>
<th>CONCEPT LEVEL</th>
<th>ARRAY</th>
<th>LEVEL</th>
<th>LATTICE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>lat</td>
<td>double</td>
<td>170</td>
<td>oph_base</td>
<td>cell</td>
<td>no</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>lon</td>
<td>double</td>
<td>180</td>
<td>oph_base</td>
<td>cell</td>
<td>no</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>double</td>
<td>24</td>
<td>oph_time</td>
<td>month</td>
<td>yes</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
The analytics framework: "metadata" operators

[37..4416] >> oph_cube
[Request]:
operator=http://127.0.0.1/ophidia/35/67;sessionid=http://127.0.0.1/ophidia/sessions/374383780832141666641463723924416/experiment;exec_mode=sync;ncore=1;cwd=/

[JobID]:
http://127.0.0.1/ophidia/sessions/374383780832141666641463723924416/experiment?82#190

[Response]:
Datacube Information
[37..4416] >> oph_cubeio
[Request]:
operator=http://127.0.0.1/ophidia/35/67;sessionid=http://127.0.0.1/ophidia/sessions/374383780832141666641463723924416/experiment;exec_mode=sync;ncore=1;cube=http://127.0.0.1/ophidia/35/74;cwd=/

[JobID]:
http://127.0.0.1/ophidia/sessions/374383780832141666641463723924416/experiment?82#176

[Response]:
Cube Provenance

INPUT CUBE | OPERATION | OUTPUT CUBE | SOURCE
-------------------|----------|-------------|-------------
http://127.0.0.1/ophidia/35/66 | ROOT | http://127.0.0.1/ophidia/35/66 | /

Cube Provenance Graph

Directed Graph DOT string:
Digraph DG {
ode [shape=box]
0 [label="PID : http://127.0.0.1/ophidia/35/74"]
1 [label="PID : http://127.0.0.1/ophidia/35/72"]
2 [label="PID : http://127.0.0.1/ophidia/35/71"]
3 [label="PID : http://127.0.0.1/ophidia/35/70"]
4 [label="PID : http://127.0.0.1/ophidia/35/66"]
5 [label="PID : http://127.0.0.1/ophidia/35/67"]
1->0 [label="oph_aggregate"]
2->1 [label="oph_merge"]
}
The analytics framework: “metadata” operators

```
[37..4416] >> oph_cubeschema cube=http://127.0.0.1/ophidia/35/67;
[Request]:
operator=oph_cubeschema;cube=http://127.0.0.1/ophidia/35/67;sessionid=http://127.0.0.1/ophidia/sessions/3743837808321466641463737283924416/experiment;exec_mode=sync;ncores=1;cwd=/;

[37..4416] >> oph_exploreCube cube=http://127.0.0.1/ophidia/35/67;subset_dim=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_dim=lat|lon|time;subset_filter=39:42|15:19|1:275;show_time=yes;sessionid=http://127.0.0.1/ophidia/sessions/743837808321466641463737283924416/experiment;exec_mode=sync;ncores=1;cwd=/;

[JobID]:
http://127.0.0.1/ophidia/sessions/3743837808321466641463737283924416/experiment?106#224

[Response]:
```
Provenance management (PID-based)
From a user experiment to a scientific workflow

- A Data Analytics Workflow Modelling Language (DAWML) has been defined
- **Extensible** schema jointly defined with application-domain scientists
- The schema allows the definition of abstract workflows

Workflow JSON representation

```json
{
    "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=$2",
                "container=tmp"
            }],
            "dependencies": ["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": ["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"
        },
            "dependencies": ["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": ["Loop for subset months"
        }
    ]
}
```
Workflow submission

```

[17, 6388] >> view 247

[Response]: Workflow Status
OPH_STATUS_COMPLETED

Workflow Progress

NUMBER OF COMPLETED TASKS | TOTAL NUMBER OF TASKS
-------------------------------
02                          | 02

Workflow Task List

<table>
<thead>
<tr>
<th>OPH JOB ID</th>
<th>SESSION CODE</th>
<th>WORKFLOW ID</th>
<th>MARKER ID</th>
<th>PARENT MARKER ID</th>
<th>TASK NAME</th>
<th>TYPE</th>
<th>EXIT STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3146">http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3146</a></td>
<td>37669923831130223251</td>
<td>247</td>
<td>3146</td>
<td>3144</td>
<td>Compute operation over time (1)</td>
<td>SIM</td>
<td>OPH_STATUS_COMPLETED</td>
</tr>
<tr>
<td><a href="http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3147">http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3147</a></td>
<td>37669923831130223251</td>
<td>247</td>
<td>3147</td>
<td>3144</td>
<td>Compute operation over time (2)</td>
<td>SIM</td>
<td>OPH_STATUS_COMPLETED</td>
</tr>
<tr>
<td><a href="http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3148">http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3148</a></td>
<td>37669923831130223251</td>
<td>247</td>
<td>3148</td>
<td>3144</td>
<td>Conversion from Kelvin to Celsius degrees (1)</td>
<td>SIM</td>
<td>OPH_STATUS_COMPLETED</td>
</tr>
<tr>
<td><a href="http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3149">http://193.284.199.174/ophidia/sessions/376699238311302232511449455166146380/experiment?247#3149</a></td>
<td>37669923831130223251</td>
<td>247</td>
<td>3149</td>
<td>3144</td>
<td>Conversion from Kelvin to Celsius degrees (2)</td>
<td>SIM</td>
<td>OPH_STATUS_COMPLETED</td>
</tr>
</tbody>
</table>
```
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)

Parallel approach
  Inter-parallelism & Intra-parallelism
Workflows in Ophidia

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 runtime execution (fire danger analysis)

https://www.youtube.com/watch?v=vxbYF1Zhpcu&feature=youtu.be
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.)

https://pypi.python.org/pypi/PyOphidia/1.2.1

https://www.youtube.com/watch?v=8pcrBXboF6U&feature=youtu.be
High Performance Data Mining & Analytics for eScience

PARALLEL
Parallel computing approach for data analytics

Scientific
Analytics framework for scientific data management

Extensible
API available to enable end-users extensions

Server-side
Remote data processing based on standard interfaces

Ophidia is a CMCC Foundation research project addressing big data challenges for eScience. It provides support for data-intensive analysis exploiting advanced parallel computing techniques and smart data distribution methods. It exploits an array-based storage model and a hierarchical storage organisation to partition and distribute multidimensional scientific datasets over multiple nodes. The Ophidia analytics framework can be exploited in different scientific domains (e.g. Climate Change, Earth Sciences, Life Sciences) and with very heterogeneous sets of data.
User, administration and devel guides
Conclusions

✓ Ophidia is a big data analytics framework for eScience

✓ OLAP approach for big data – multidimensional data model

✓ Multiple use cases for data analysis in different domains/contexts have been implemented

✓ It provides access via CLI (end-users) and API (devel users)

✓ Programmatic access via C and Python APIs

✓ Several deployment scenarios in the cloud have been implemented
  ✓ Mainly in the EUBrazilCC, CLIPC and INDIGO project

✓ Official Release available from February 1st 2016 on github
Ophidia – Useful Resources

• Website: http://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
Thanks

http://ophidia.cmcc.it

@OphidiaBigData

www.youtube.com/user/OphidiaBigData