INDIGO DataCloud Tutorial: testing Ophidia on an astronomical image calibration task

Elisa Londero - londero@oats.inaf.it

INAF - Trieste Astronomical Observatory (OATs), 34143 Trieste, Italy

Dear User,
in order to test Ophidia for the reduction of astronomical images you are kindly requested to follow the procedure listed below. Before you start you should have already registered a new OphidiaLab account and have received your credentials via email. If not, please register here.

This tutorial presents the outcome of a use-case developed within a collaboration between INAF (Trieste) and CMCC (Lecce) in the context of the INDIGO DataCloud European project for the development and improvement of open-source cloud-based software.

Ophidia is a framework for data intensive analysis that exploits advanced parallel computing techniques and smart data distribution methods. In particular, the point of strength that makes it extremely efficient in parallel data processing is the adoption of an array-based storage model and a hierarchical data organization for the distribution of scientific datasets over multiple nodes. The framework was developed in the context of climate change and it has been mostly used in that field up to now. With this use-case we extend Ophidia's capabilities to the field of astronomy. Ophidia has been developed within the CMCC Foundation. More information about the project can be found here: `http://ophidia.cmcc.it/`.

OphidiaLab is a science gateway developed by the CMCC Foundation that offers to the end-user the possibility to exploit the full Ophidia potential directly from the browser. It integrates basically two components in order to create a ready-to-be-used scientific environment: a Jupiter-Hub instance and an Ophidia cluster. JupiterHub can be used to serve notebooks to scientific research users and it comes together with a large set of pre-installed Python libraries (PyOphidia module) for manipulating, analyzing and visualizing data. Jupiter Notebooks are in practice used to create and share documents containing live code. OphidiaLab gives also the possibility to use the Ophidia Terminal which is basically an Ophidia client with a functionality similar to that of the bash shell. Both Jupiter Notebooks and the Ophidia Terminal will be explored during the tutorial.

The image calibration test is based on a set of calibration images (bias and flats) and science images that are already loaded onto OphidiaLab. If you want to reduce your own set of images please send an email to `londero@oats.inaf.it`.

Thanks to the code extension developed in the context of this use-case, Ophidia can now read the FITS format. The latest release of the software (v1.1.0) includes this feature. If you are interested to install it locally on your machine, please download the code and install it according to the procedure described here.

Once loaded onto Ophidia, the FITS files are identified by two metadata keys: the filter type (FLT_ID) and the observation type (OBS_TYPE). In the set of images specifically used for this tutorial, three different filters are found: B_JOHN_10, V_JOHN_11 and R_JOHN_12. The observation type specifies if an image is of bias, flat or science type. In case you want to reduce your own set of images you must know the values of these two metadata keys in advance because they need to be set in the Python script.

The tutorial begins here.

- **Login to OphidiaLab**

  1. By using the credentials received by email, login to OphidiaLab following this link.
  2. Get acquainted with the environment: you can find a short guide explaining OphidiaLab capabilities in your home folder under quickstart → Quick_Start.ipynb
  3. The folder INDIGO-FITS/fits-raw in your home contains all the FITS files that are going to be used for this calibration example.

- **The Ophidia Terminal**

  1. Open an Ophidia Terminal by clicking on the *New* button on the top-right part of your home page and select *Terminal*.
  2. Use the `pdd` command to check your current directory.
  3. Use the `lsd` command to list directories and files.
  4. Create the empty container called FITS2:

     ```
     oph_createcontainer container=FITS2;dim=NAXIS2|NAXIS1;dim_type=int|int;
     ```

  5. Load a FITS file (from the already available repository) in the container by giving the command:

     ```
     oph_importfits src_path=[path=/data/INDIGO-FITS/fits-raw;
     file=LRS.2016-06-06T20-30-03.925.fits];measure=mymeasure;
     exp_dim=NAXIS1;imp_dim=NAXIS2;container=FITS2
     ```

  6. Use the command `ll` to check that the cube has been loaded.
  7. In order to check that all the metadata have been loaded, replace the stars with the PID number identifying your cube in the following command (see picture):

     ```
     oph_metadata mode=read;cube=https://ophidialab.cmcc.it/ophidia/***/***;
     ```

8. To select the metadata value corresponding to the FLT_ID key replace the stars with the PID number identifying your cube as before:

```
oph_metadata mode=read;cube=https://ophidialab.cmcc.it/ophidia/***/***;
metadata_key=FLT_ID;metadata_type=text;metadata_value=test_text;
```

9. Delete the cube by issuing this command, where again the stars need to be replaced:

```
oph_delete cube=https://ophidialab.cmcc.it/ophidia/***/***;
```

10. If you experience problems try to remove the FITS2 container by using:

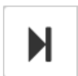
```
oph_deletecontainer container=FITS2;
```

and restart from scratch.

This section has shown that it is possible to execute Ophidia operators to manipulate the FITS files directly from the Ophidia terminal. From the same terminal it is also possible to submit workflows specified in a JSON file. If you want to learn more about the commands available for the Ophidia Terminal you can check this page.

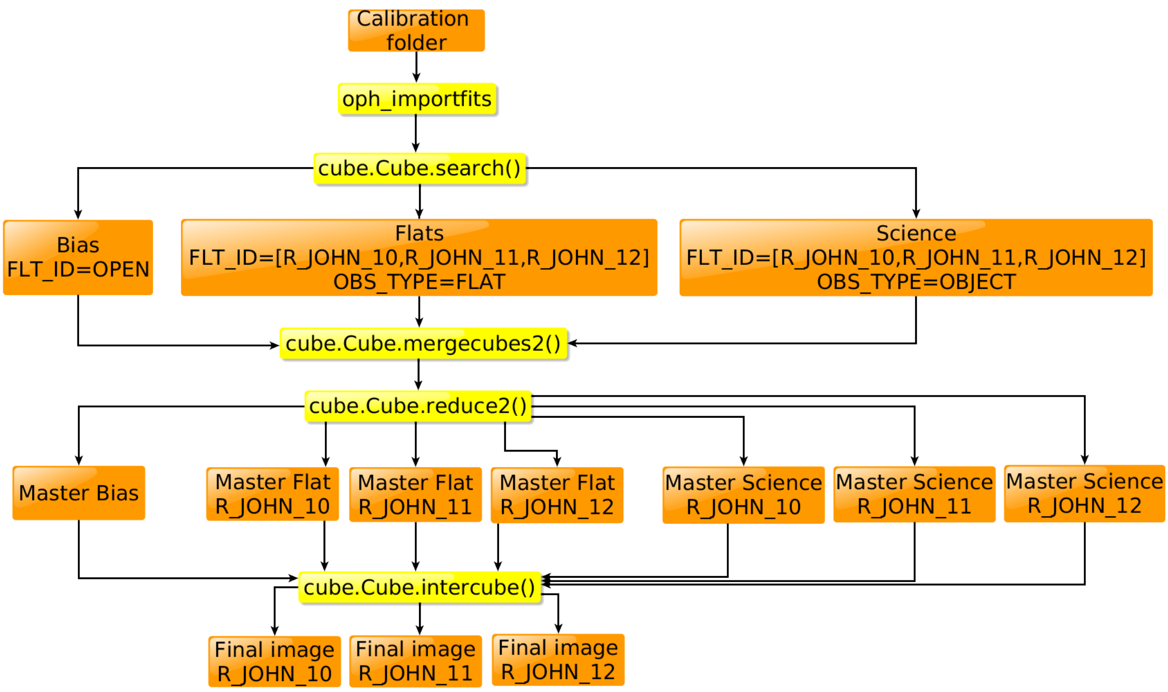
- **Run the calibration script from a Jupiter Notebook**

1. Now we start to explore the Jupiter environment. A reduction Python script has already been written for you. Download it locally on your machine from here.
2. Click on the *Upload* button on the top-right part of your home page and search for the file you want to upload.
3. Click on the blue *Upload* button that it appears in your home; after that you should see the file among the Notebooks in your home.
4. Open it by clicking on it. The file is divided in Sections.
5. In Section I fill in the credentials used for accessing OphidiaLab; this way you will connect to the server instance. Notice that from PyOphidia the modules *client* and *cube* are imported: the first is used for submitting any type of request (from simple tasks to workflows) while the second makes the direct interaction with the cubes (in our case the metadata and data of the FITS files) possible. Click on the cell and run it by using the run cell button at the top of the page identified by the symbol:



6. In Section II the FITS files are imported onto Ophidia by specifying the path and the dimensions of the files (NAXIS1 and NAXIS2 for optical images). Click on the cell and run it. The following two cells set the values of the metadata for the filters used and for the observation type (flat, science or bias). Run both the cells. If you are analyzing your own set of images, set the values for the filters here.
7. In Section III the median of all the bias files is calculated; run the cell.
8. Sections IV-VI: the median for the flat files is set-up without doing the calculation. Each cell performs the operation for one particular filter. Run the cells.
9. Sections VII-IX: the median for the science images is set-up without doing the calculation. Each of the three cells performs the operation for a different filter. Run all the cells.

10. Sections X and XI calculate the median for the flat and science images respectively. Run all the cells.
11. In Section XII the bias image obtained in Section III is subtracted from the three science images resulting from Sections VII-IX. Run the cell.
12. In Section XIII the bias image obtained in Section III is subtracted from the three flat images resulting from Sections IV-VI. Run the cell.
13. In Section XIV the three science images obtained in Section XII are divided by the flat images obtained in Section XIII. Run the cell.
14. Finally the result of the calibration procedure is written to a FITS file by using the oph_exportfits Ophidia operator. Replace the stars with your username. Run the cell.
15. Now you can download the file and visualize it by using for example fv. The figure below shows a summary of the procedure that has been followed.
16. In case you get stuck and want to run all the cells again, first remove all the cubes by using the Ophidia Terminal. Do the following: first issue `ll` to check the status of the loaded cubes; then remove all the cubes by the command `oph_delete cube=[***:***]`. The stars have to be replaced by the last number characterizing a cube PID.



This section has shown the structure of the calibration script and the flexibility of the Jupiter Notebooks. The script can be easily adapted to other calibration tasks according to the needs of the user.

- **Considerations on the performance**
  There are 100 cores all together on the CMCC cluster and the script you have tested uses 2 cores for each task. In order to improve the performance you can increase the number of cores and measure the time elapsed by using a structure of this type:

```
import time

start = time.time()
```

```
end = time.time()
print(end - start)
```

Remember that there is a number of users logged into the system so the performance can vary depending on the load on the system.

- **Conclusions and final remarks**
  This tutorial has shown how to use Ophidia for an astronomical image calibration task. In order to make Ophidia competitive with other already available software and to extend its use also to other astronomical analysis tasks, further development of the code is needed. To this end it is important to us if you could express your degree of satisfaction with this tutorial and we would also like to know your opinion on a possible future extension of the Ophidia capabilities to the astronomical field. Would it be valuable for your research? The **survey** you are asked to fill in can be found following this link. Thank you for your cooperation.