Web-based interactive analysis and processing of Earth Observation data at petabyte scale

European Commission, Joint Research Centre
Directorate I Competences, Unit I.3 Text and Data Mining

Contacts: Armin.Burger@jrc.ec.europa.eu
Pierre.Soille@jrc.ec.europa.eu
Earth Observation & Social Sensing
Big Data Pilot Project - Background

- Wide usage of Earth Observation (EO) data at JRC
- The EU **Copernicus** Programme with the **Sentinel** fleet of satellites acts as a game changer:
  - expected 10TB/day of **free and open data**
  - Requires new approaches for data management and processing
- JRC Pilot project launched in 2015
  Major goal: set up a common platform
  ⇒ **JRC Earth Observation Data and Processing Platform (JEODPP)**
JRC Earth Observation Data Processing Platform (JEODPP)

- Main focus on *satellite image* data
- Support for existing processing workflows and environments (C/C++, Python, Matlab, Java)
- Development of processing API for EO data
  - *in Python, based on C/C++ modules*
  - *used for*
    - Low-level batch processing
    - High-level interactive processing
Components of the **JEODPP**

**Batch processing**

**Interactive processing**

**Distributed file system**

**High performance network**

**Computing Clients**
Low-level batch processing

- Running large-scale data processing tasks in a cluster environment
- HTCondor workload manager
  - Using **Docker** containers for executing jobs
    => flexible management of processing environments
  - Initial processing runs are very promising:
    - Good scalability
    - High data throughput in combination with storage system
High-level interactive visualization and processing

- Web user interface to server-based data processing
  - Based on Jupyter Notebook, run inside Docker container
  - Display processing results on an interactive map
  - Split processing to multiple parallel tasks via the interactive map (deferred execution)

- First prototype implemented

- On-going
  - Performance optimization
  - Resource management in multi-user environment
  - Interaction with workload manager
  - Binding with JRC and other image processing libraries
Interactive visualization example

```python
from jiplib import inter
from jiplib.map import Map

map = Map()
map

In [5]:
coll = inter.ImageCollection("S2")
coll = coll.FilterOnGeoName("Urbania")
coll = coll.FilterOnDate(2015,1,1, 2016,10,18)
coll = coll.FilterOn("cloudcover", "<=", 15)
coll = coll.FilterOn("jrc_filepath", "<>", "")
coll.Limit(10)
coll.Sort("cloudcover", True)

imgList = coll.List().Bands("B04","B03","B02",1.5).Shift(0)
map.zoomToImageExtent(imgList)
map.layer = map.addLayer(imgList.toLayer(), nocache=True)
```
Interactive processing – set-up

Interactive Image Processing in JEODPP

**Jupyter notebook**

```python
coll = ImageCollection("S2").filterGeo("Ispra")
coll = coll.filterOnDate("2016,1,1, 2016,12,31")
coll = coll.sort("beginposition").limit(10)
proc = coll.process().band("B04")...
m = Map()
m.addLayer(proc)
```

**ipyLeaflet interactive map**

**Database**
- EO Metadata
- Processing chains

**EOS storage**
- Read images from storage using overviews

**Load balanced web gateway**

**Jiplib**
- Interactive Image library

**Filter images on metadata + Stores processing chains**

**Call Jiplib**

**Emits HTML and javascript code**

**Request for tiles download**

**Send back tiles**
Interactive processing demo

>>> Start Notebook Application Hub
Sharing data processing chains via Jupyter Notebooks

- Share processing chain definition instead of data from processing results
- Save, exchange, and load Notebook files
  - display results of processing chain from other users in your own Jupyter Notebook
  - adapt processing chain
  - caching mechanisms could be used to speed up display of results
Sharing processed data via Web services for selected products

e.g. via standardized Web Map Service (WMS) protocol

Example:
Global Human Settlement Layer (GHSL) derived from Copernicus Sentinel-1 data

Service loaded in a desktop application

Service URL: http://cidportal.jrc.ec.europa.eu/jeodpp/services/ows/wms/landcover/ghsl?
Sharing processed data via Web services for selected products

JRC Global Human Settlement Layer (GHSL) Web Map Service, loaded in a web mapping application

In testing: data sharing and terminal services

- NextCloud instance
  - *For user Notebooks, user Python modules, shared data,...*

- Terminal services via Web interface
  - *Desktop data analysis tools, e.g. Matlab, ENVI, R, QGIS,... inside processing infrastructure*
  - *User data folders from NextCloud instance mounted*
Thank you for your attention

Zoom on Sentinel 2A tile 33UWP maximum distance to cloud composition with data from 26/04/16 to 16/05/2016