“Big Data and Earth Observation”

https://github.com/fchouteau/isae-otsu

About

Florient CHOUTEAU

ISAE-SUPAERO 2016, Filière SDD

Machine Learning Engineer at Airbus Defence and Space (Space Systems)

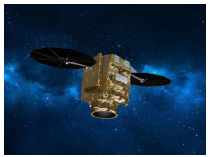
Working in EO since 2016 (PFE) at Delair, Geo-Intelligence, then Airbus

AIRBUS

Disclaimer : I am here on my own time (holidays)

Airbus & Earth Observation

- Airbus Defence and Space builds space/ground systems for Earth Observation
- Some examples (optical)
  - SPOT (Satellite pour l'Observation de la Terre)
  - Pléiades Haute-Resolution with Thales Alenia Space, for CNES
  - Pleiades Neo
  - CO3D "Constellation Optique 3D"



- Airbus Defence and Space operates satellites and sell imagery & services

Context

Class “Bases de données et apprentissage”

- Cloud computing, Virtual Machines & GCP
- Deep Learning, Image Segmentation with U-Nets

01/02 : Class + Hands-On

- Introduction to “Big Data” Challenges in EO
- From Pixels to Jupyter Notebooks
- Introduction to distributed computing principles

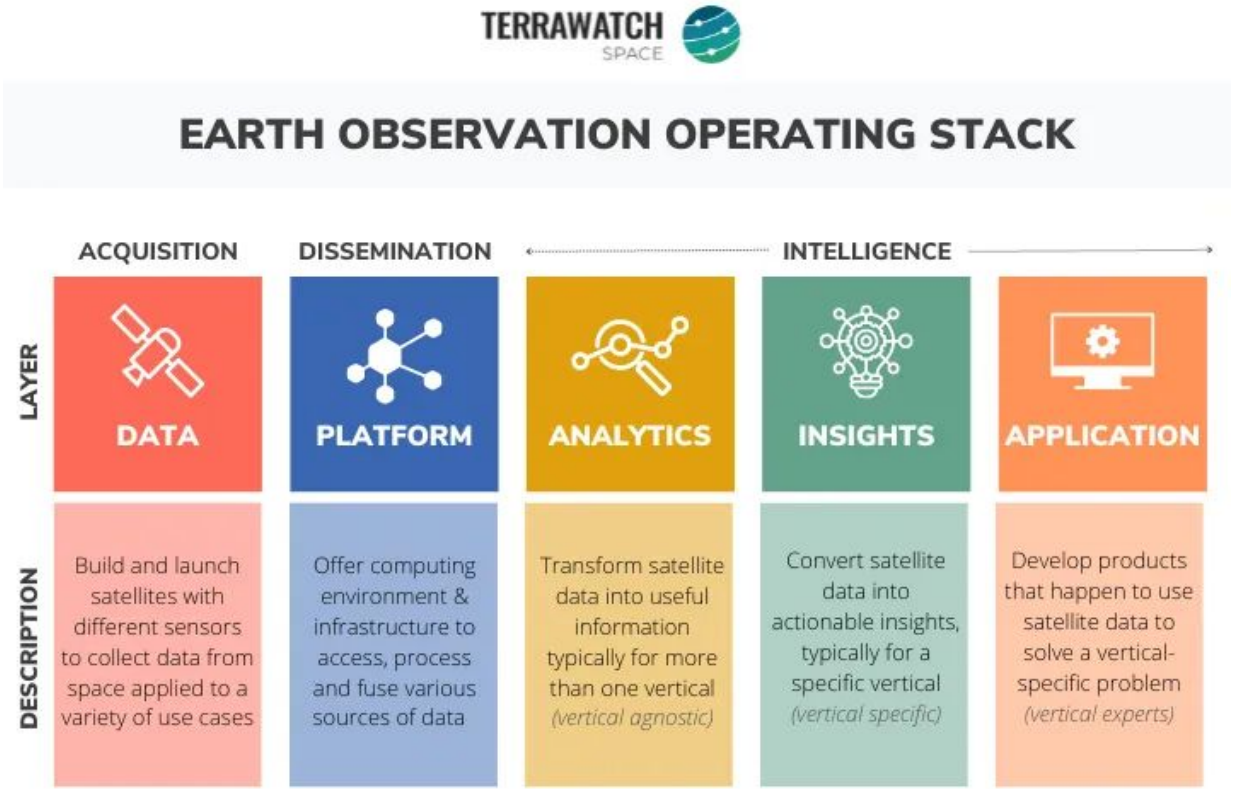
07/02 : BE

- Distributed computing and Dask for (massive) data processing

1 - “Big Data” in EO

Introduction to some challenges

The Earth Observation “Operating Stack”




Pléiades Neo Satellite image - Toulouse, France © Airbus DS 2022

Examples : Pléiades Neo

https://www.intelligence-airbusds.com/en/5750-image-gallery-results?search=gallery&world=&market=&continent=&sensor=4753&submit=

Examples : Pléiades Neo

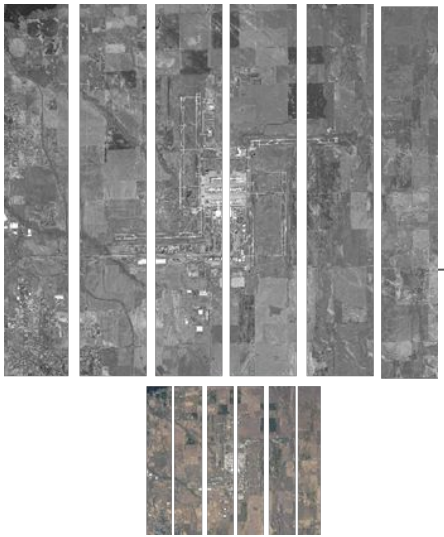
- 2 satellites acquiring at 30cm GSD
- 14km swath, 47k x 47k pixels for a “scene”

Acquisition : Volumetry

- Each Pleiades Neo acquires 500k km² / day / satellite
  - 1.2 GB/s downlink rate
- 2500 scenes / day / satellite
  - Ground segment must process 5000 scenes / day
  - At least archiving
- 912 500 scenes / year / satellite

Dissemination : Storage

- Need to produce the images
- A compressed 14 x 14 km Pléiades Neo product weights 7.3 Gb (compressed JP2)
  - 36.5 Tb / day of storage for produced data



- Despatialise
- Primary Sensor
- Orthorectify

Analytics : Computing time

- How to run your U-Net on all cities of the earth ?

- Let’s say 2 Mpix / s (1024 x 1024 pixels) on a state of the art GPU
  - 2100 Mpix for one scene 14 x 14 km
  - 18 minutes processing time per scene
  - 300h our computing time per day to process 1000 scenes / day (20% of the acquisition)

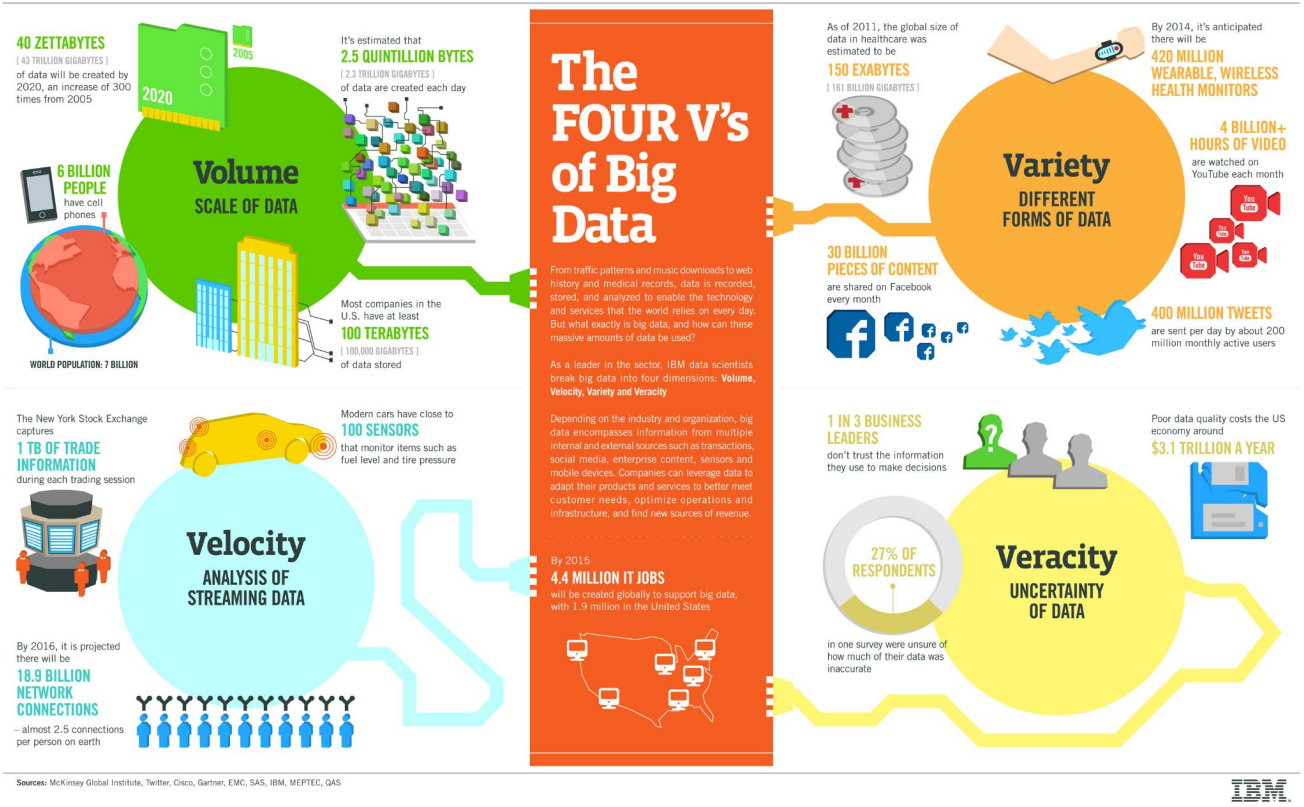
It’s only 1 analytics on 20% of the data

This is only an example…

- Planet currently has 212 (tbc) SuperDoves (3.7m GSD) in orbit
- Planet plans to launch 32 Pelicans with 30cm GSD
- Sentinel 2A/B/C are acquiring the earth every 5 days at 10m GSD

You can find examples in astrophysics as well : Gaïa, LSST, …

“Big Data”


The FOUR V's of Big Data
From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?
As a leader in the sector, IBM data scientists break big data into four dimensions: Volume, Velocity, Variety and Veracity
Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.
By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States
Volume
SCALE OF DATA
40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005
2005
2020
6 BILLION PEOPLE
have cell phones
WORLD POPULATION: 7 BILLION
It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day
Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored
Variety
DIFFERENT FORMS OF DATA
As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]
By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS
4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month
30 BILLION PIECES OF CONTENT
are shared on Facebook every month
400 MILLION TWEETS
are sent per day by about 200 million monthly active users
Velocity
ANALYSIS OF STREAMING DATA
The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session
Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth
Veracity
UNCERTAINTY OF DATA
1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions
27% OF RESPONDENTS
in one survey were unsure of how much of their data was inaccurate
Poor data quality costs the US economy around
$3.1 TRILLION A YEAR
Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS
IBM.

“Big Data”

- Computing platforms
  - High Performance Computing / Clusters
- Tools
  - Data processing
  - Data manipulation
  - Data visualisation
- Mindset
  - “Cloud Native”
  - Automated analysis tools

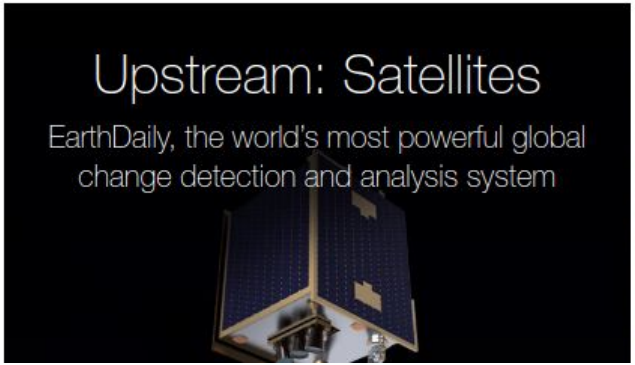
More about (real) big data

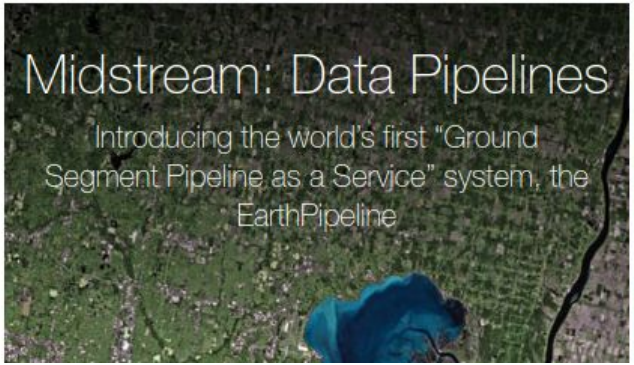
- https://cnes.github.io/big-data-processing-course/

2 - Observing the Earth from your Jupyter Notebook

How do we do to make sense of such a volumetry of data ?

Upstream, Midstream, Downstream





The problem with EO

1. Data Availability
   Acquiring, processing the data

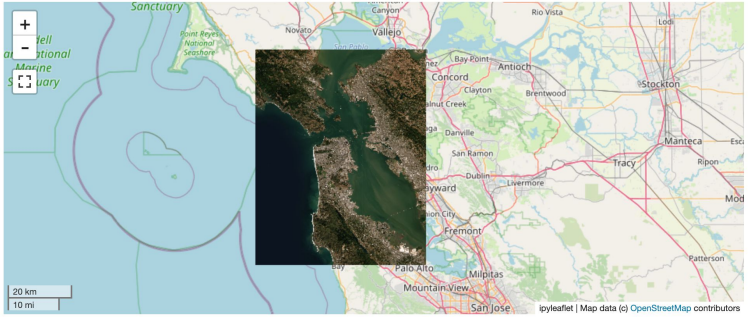
2. Data Accessibility
   Being able to actually access the data

3. Data “Fusability”
   Working with different sources of EO data

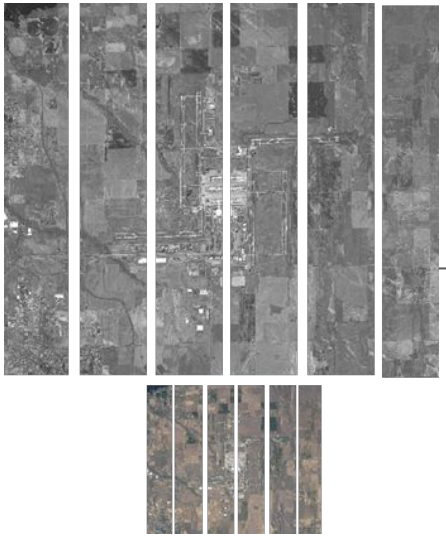
4. Data Usability
   Using the data to do actually useful things

Objectives

- Recap of the journey of an image from the sensor to your notebook
- How we leverage the cloud to enable access to EO data
- The problem with referencing & an introduction to STAC
- The problem with storage & an introduction to COG
- The problem with access / manipulation & an introduction to modern tools



The first steps of the journey



- Despatialise
- Primary Sensor
- Orthorectify

What’s next ?

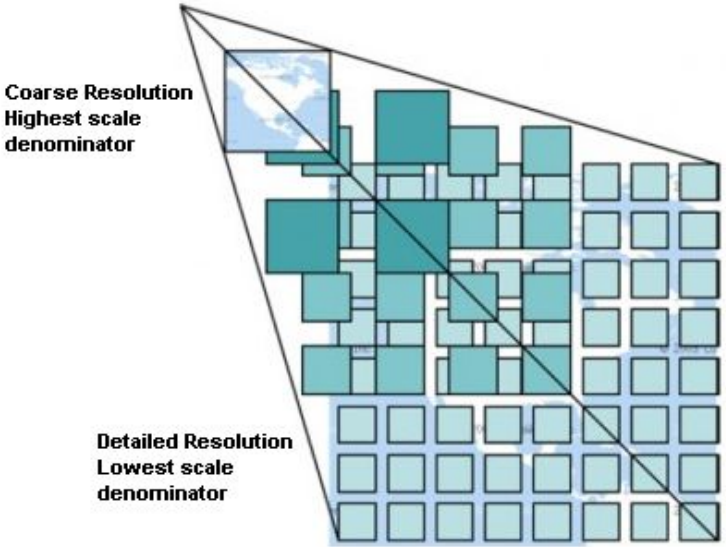
Leveraging the cloud

- We can dump all of our images on a storage
- How do we reference the data ?
- How do we allow easy access to users ?

Storing data so that people can actually access it

- A Pleiades Neo Image of 14 x 14km : 25 Gb
- Do we download the 25 Gb image each time we want to look at something ?
- We don’t have that much storage in our computers !

Enter... tiles !



COG

CLOUD OPTIMIZED

GEOTIFF

This powers a lot of applications

https://geojson.io/#map=2/20.0/0.0

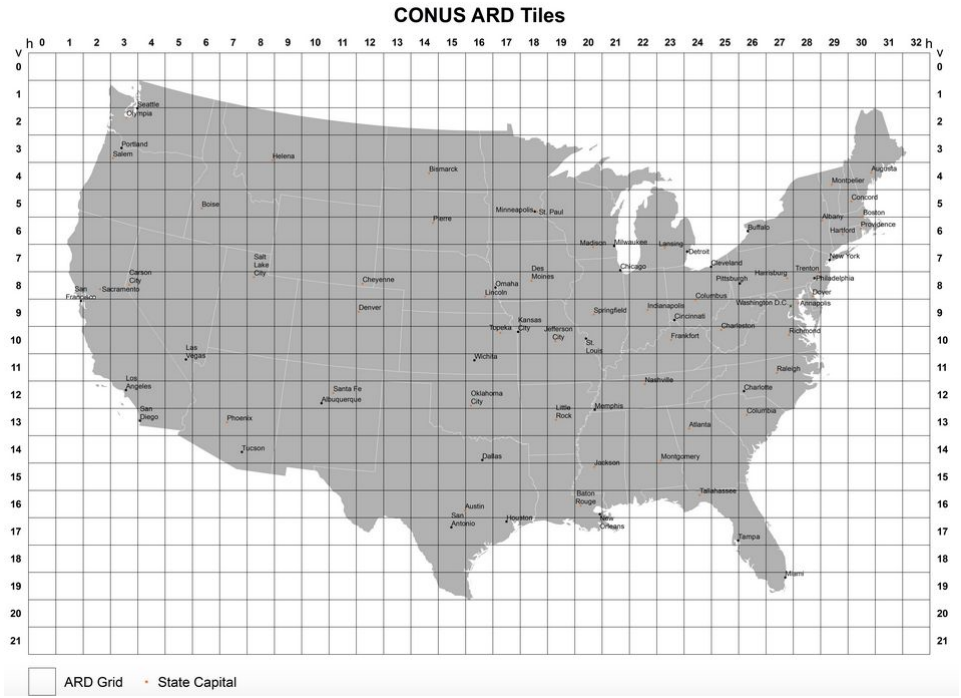
https://www.google.fr/maps

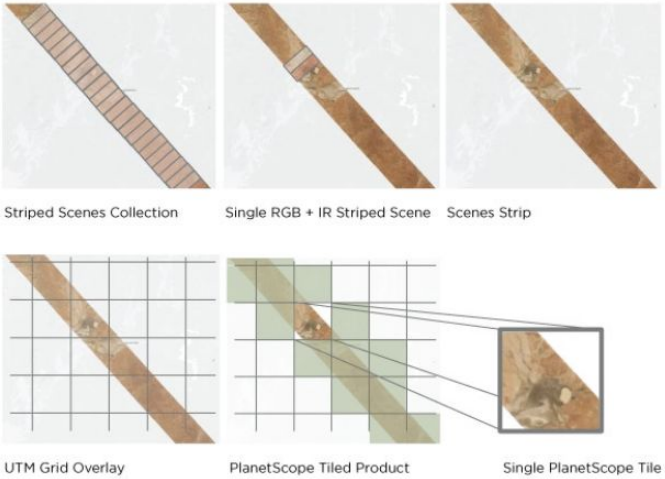
Referencing Data

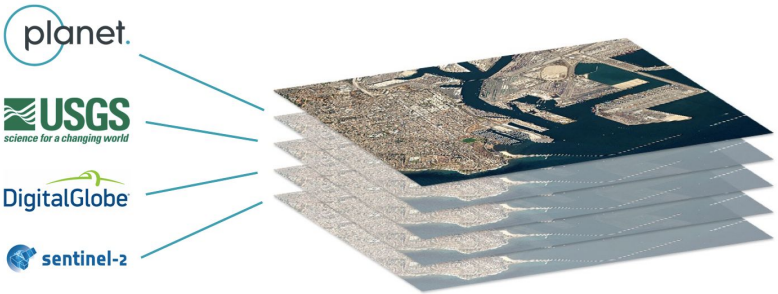
https://stacindex.org/catalogs/planet-labs-stac-catalog

Fusing Data : Objective

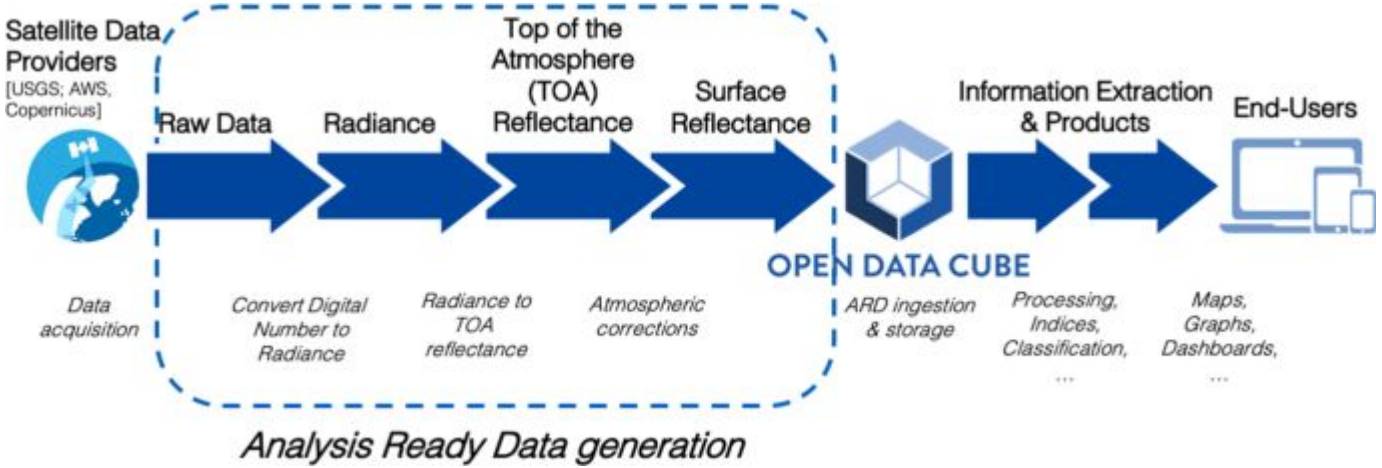
Fusing Data : Geometry







Fusing Data : Processing Level



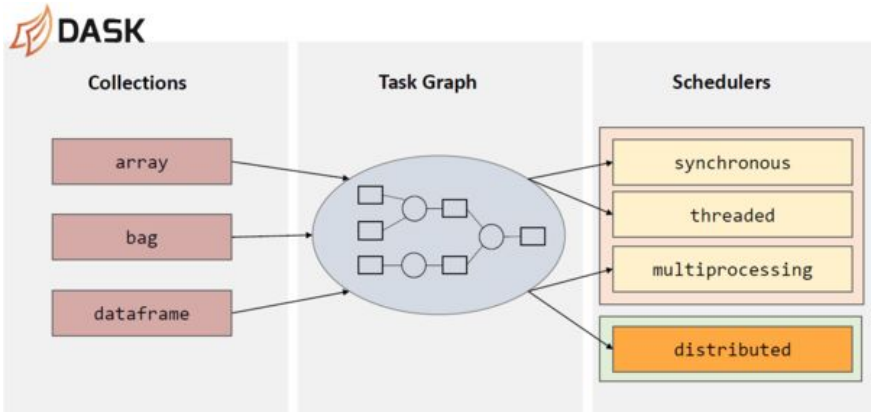
Fusing Data

https://medium.com/planet-stories/analysis-ready-data-defined-5694f6f48815

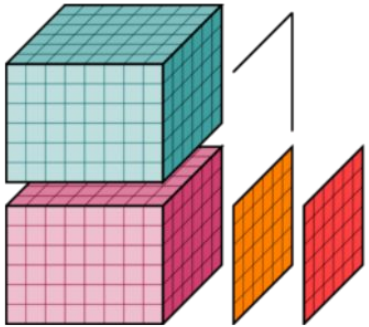
https://www.maxar.com/products/analysis-ready-data

Using data : Some technologies

- Partial Read
- Lazy Computation



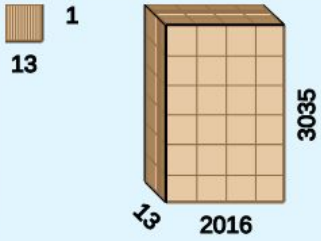
Using data : Some technologies



xarray

xarray.DataArray (**time**: 13, **band**: 13, **y**: 3035, **x**: 2016)

	Array	Chunk
Bytes	4.14 GB	1.05 MB
Shape	(13, 13, 3035, 2016)	(1, 1, 512, 512)
Count	16393 Tasks	4056 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

band	(band)	object	'B01' 'B02' 'B03' ... 'B8A' 'CLD'
y	(y)	float64	0.5 1.5 2.5 ... 3.034e+03 3.034e+03
x	(x)	float64	0.5 1.5 2.5 ... 2.014e+03 2.016e+03
time	(time)	datetime64[ns]	2019-06-06 ... 2019-11-03

▸ Attributes: (11)

Viewing Data

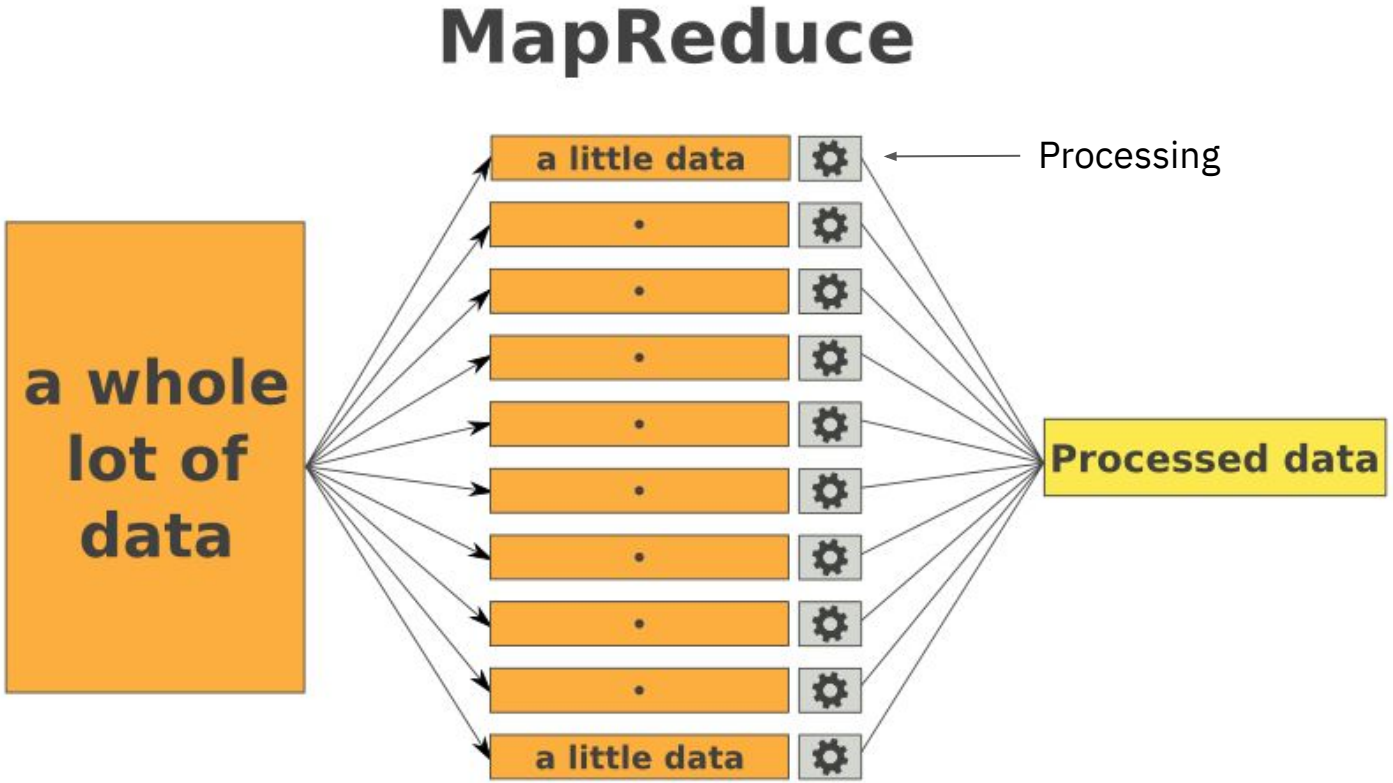
https://apps.sentinel-hub.com/eo-browser/

https://twitter.com/MykolaKozyr/status/1488574117863641091

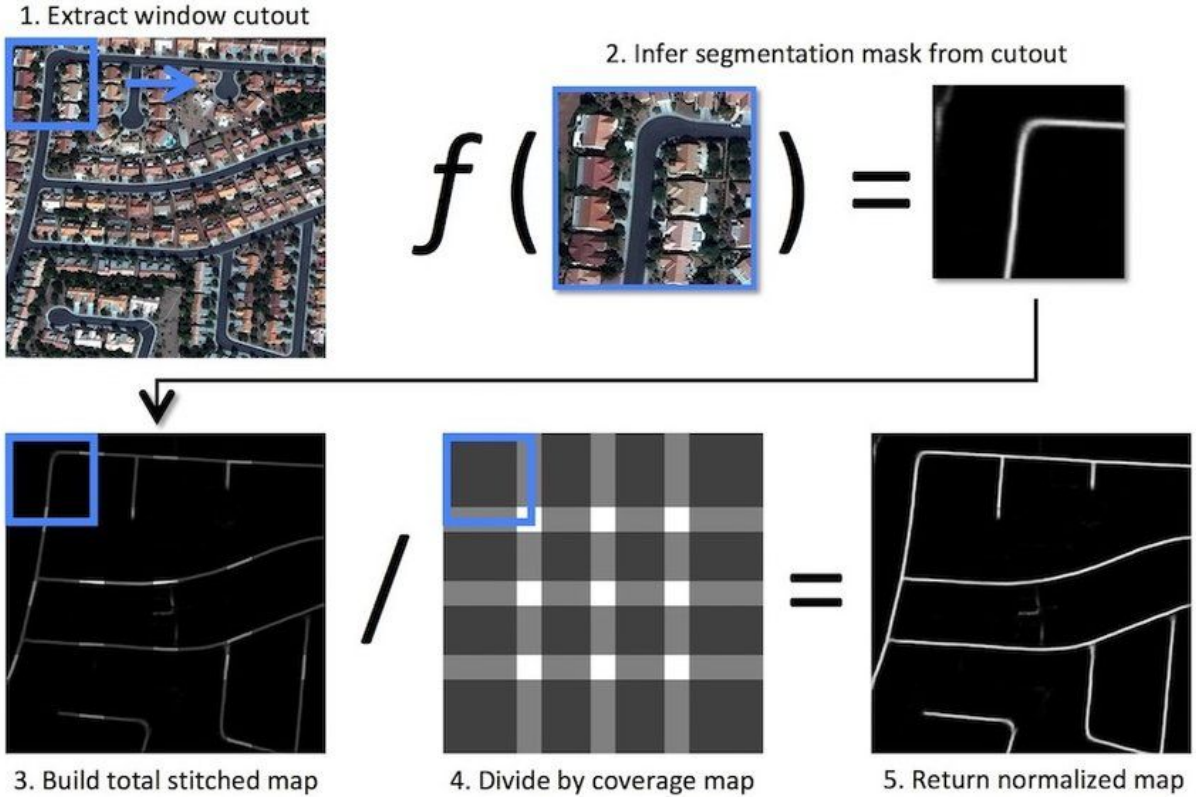
3 - Distributed Computing

Introduction & mindset

The mindset


MapReduce
a whole lot of data
a little data
a little data
Processing
Processed data

Example with satellite imagery



In python

```python
data = [0,1,2,3]
results = []

def f(x):
    return x+1

for x in data:
  y = f(x)

 results.append(y)
```

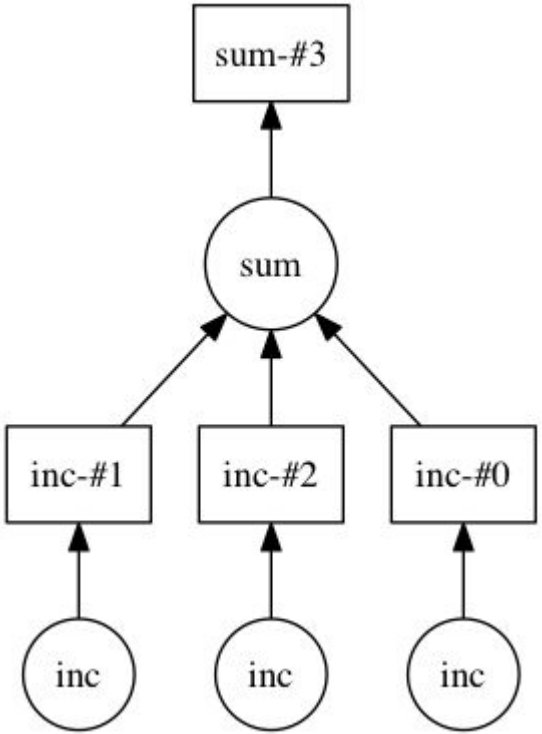


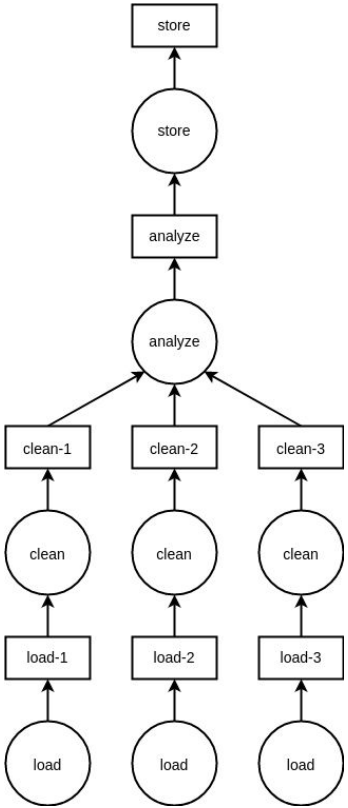
```python
data = [0,1,2,3]


def f(x):
    return x+1

results = map(f, data)
```

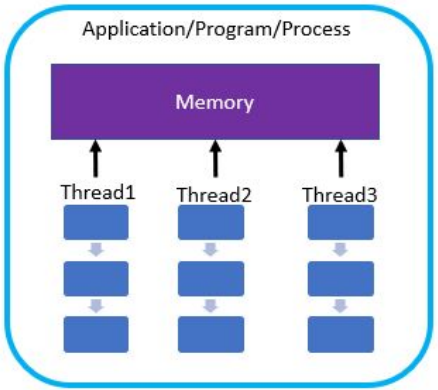
Pipelines and tasks



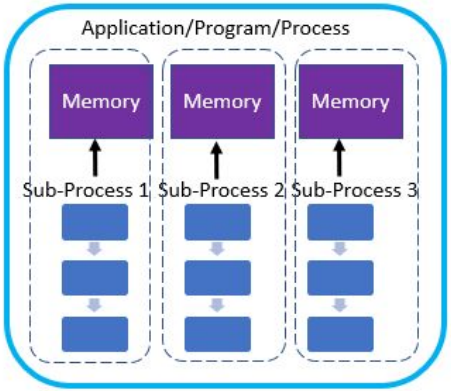


Data distribution

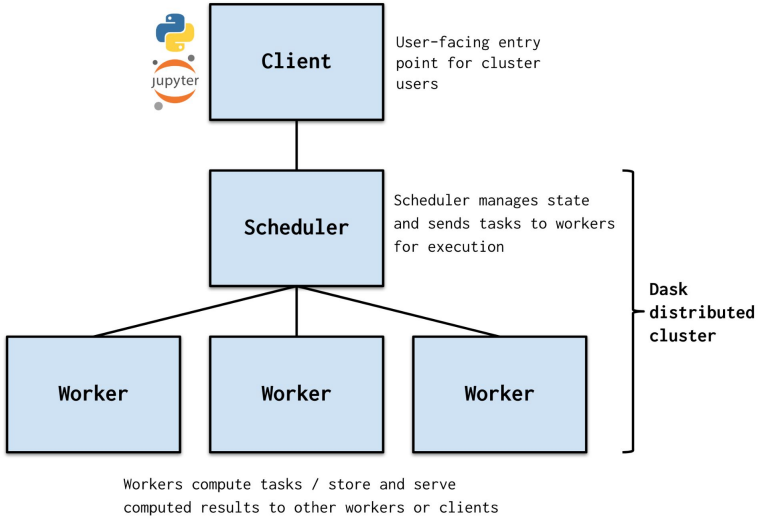
- Sur une même machine (multiprocessing)
- Sur plusieurs machines



Multi-Threading



Multi-processing



Things to think about

I want to process a huge number of satellite images

- How do I formalize all my processings as tasks ?
- How do I distribute my tasks ?
  - Per product ?
  - Per tile ?
- What are my bottlenecks?
  - Processing ?
  - I/O ?
  - Network ?

To be continued…

BE :

- Distributed data computation
- Map reduce concept in python and joblib
- Introduction to Dask

4 - Hands-On

https://github.com/fchouteau/isae-otsu