LABORATOIRE
D'OCEANOGRAPHIE
SPATIALE

Data intensive science platform - Use cases -

Benefits from Big-data and Cloud technologies

Cargo Day n°5 - Rennes - 19 Novembre 2015

Frédéric Paul (fpaul@ifremer.fr), Olivier Archer, Jean-Francois Piollé, Bertrand Chapron
Institut Français de Recherche pour l'Exploitation de la Mer (Ifremer), Brest, France

Why are we here ?

- Cargo Day, 5th edition

  “ Pour cette journée, nous allons rester très pragmatique et essayer de couvrir ce sujet vaste sur les données, mais tout en restant focalisé sur le comment notre métier et nos expériences du terrain ont su répondre et vont répondre demain à cette augmentation drastique des données de plus en plus massives que nous manipulons quotidiennement, tant sur le traitement que sur l'analyse ”

- This lightning talk goal :

  HIGHLIGHT EXAMPLE OF DATA USAGES
  NOT REALLY FEASIBLE BEFORE THE DISTRIBUTED PLATFORM

ESA's BULK REPROCESSING CAMPAIGNS

Reaper

Reprocessing of Altimeter Products for ERS-1 & ERS-2 (1991 - 2003)

- consolidation of **ERS RA and MWR** datasets
- Improved ERS orbit data for entire period
- **Homogeneous dataset**, compatible with Envisat
- Cross-calibrated between ERS-1, ERS-2, Envisat
- **Improved accuracy**

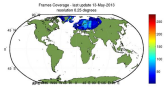
Virtual environment allows very fast and smooth integration by offering an environment cloned from REAPER processor designers

Complete ERS-1 & ERS-2 reprocessing in one week

X-PReSS

eXpert Product Reprocessing Scalable Service

Landsat 5 TM (Kiruna)
collection, consolidation, processing and repatriation



200 TB of data :
- 27576 L5 TM WILMA L0
- 150.000 ortho-rectified L1 scenes

Complete reprocessing in 2 weeks

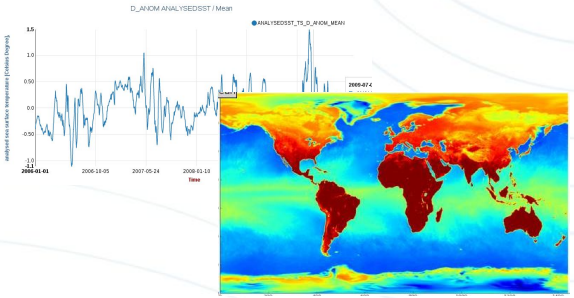
DSI

Data Consolidation and Bulk Processing Service Initiative

heavily relies on the previous experiences, to continue consolidation and processing activities of several ESA's unique legacy data holdings.

Projects ongoing :
- Envisat MIPAS reprocessing
- bulk-processing of ESA's Landsat TM, MSS and ETM+ archives

DATACRUNCH using HADOOP Map/Reduce



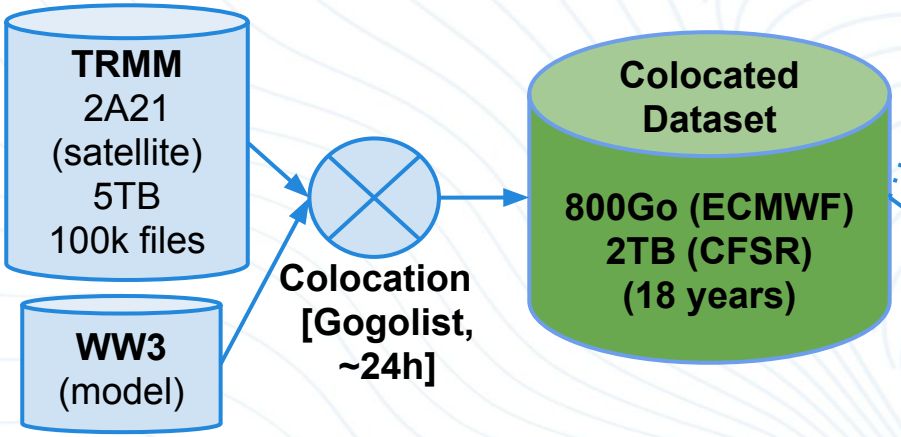
Why HADOOP for satellite data analytics ?

- MapReduce : **easy to merge datasets and extract metrics**
- Scalable BigData distributed processings : **high performances**
- Fully Fault-tolerant (hardware, software) : **easy to manage**

“DataCrunch”, a Cersat Dataset Analytics tool :

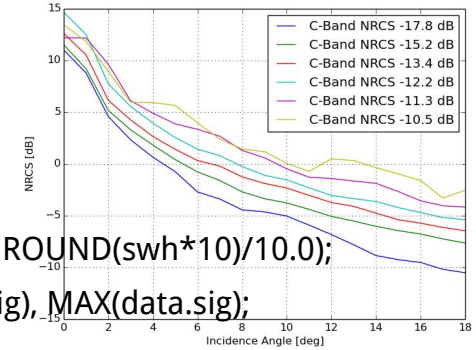
- Output : global and regional statistics
  time-series, maps, climatology, hovmoller, …
- Inputs : datasets available on the Cersat Cloud
  - Format : native (NetCDF, HDF, ...)
  - Data : satellite (L0 to L4, swath or gridded), models, buoys…
  - Size : a few MB to several TB
  - Time range : a few days to >20 years
- Adding a new dataset : <50 lines of Python or Matlab
- Usual Processing time : a [quick] coffee

Data Analytics on TB datasets using PIG within minutes



Dataset == CSV file, containing variables :
TRMM(lat,lon,incidence,sigma0, ...)
MODEL(lat, lon, wind_speed, wind_direction, swh, …)

GOAL : for each tuple (incidence, wind_speed, swh), compute AVG(sig0), NBR(sig0), MAX(sig0)



PIG CODE TO ACHIEVE GOAL :

```
data = LOAD 'coloc_2A21_v2/*.dat' USING PigStorage(' ') AS (
        lat:float, lon:float, incidence:float, sig0:float, …
        lat_ww3:flat, lon_ww3:float, wind_speed:float,swh:float, …)
tuple = GROUP data BY (ROUND(wind_speed*10)/10.0, ROUND(incidence*10)/10.0, ROUND(swh*10)/10.0);
RESULT = FOREACH tuple GENERATE FLATTEN(group), AVG(data.sig), COUNT(data.sig), MAX(data.sig);
STORE D INTO 'myresults.txt';
```

3min + 4 lines of PIG code ...

Data Analytics on TB datasets using Spark within minutes

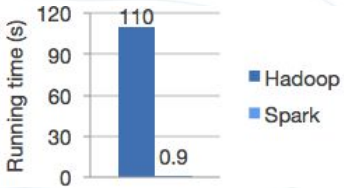
Fast and general engine for large-scale data processing.

Same exercise as Pig, but using Spark as map/reduce engine

- More code than PIG, but not much
- Faster (aggressive memory usage)
- More convenient for interactive analysis
- Suitable for iterative algorithms
- (Less robust than Hadoop M/R ..)

Ideal when the dataset you want to analyse can be stored in distributed memory (~TB order of magnitude) ?

<u>Really promising !</u>




```
#!/opt/shark/spark-0.8.0/pyspark

#setenv MASTER=spark://skyfall:7077 ; setenv SPARK_MEM 8g ; ./pyspark

def make_kv(csvline):
    res = [ float(n.strip()) for n in csvline.split(' ') ]
    spd_wind, dir, zon_wind, mer_wind, lon_wind, latitude_wind, inc, sig,\
            scOrientation, lon, lat, spd_wind_ww3_ravel, dir_wind_ww3, \
            zon_wind_ww3, mer_wind_ww3, lon_ww3, lat_ww3, swh_ww3 = res
    return (round(spd_wind,1), round(inc,1), round(swh_ww3,1)), (sig, 1)

def reducer(x, y):
    return (x[0]+y[0], x[1]+y[1])

def computemean(x):
    return  float(x[0])/x[1]

def filter_wind(x):
    if 12 > x[0][0] > 10: return True
    else: return False


# Load colocated dataset from files
dat = sc.textFile('hdfs://br156-244//user/fpaul/coloc_2A21_v2/coloc_2A21.2014*')
# Store in distributed memory K/V : (windspeed, incidence, swh) -> (sig0, 1)
datcache = dat.map(make_kv).cache()

# Compute Sig0 mean for each tuple (windspeed, incidence, swh)
sig0mean_cache = datcache.reduceByKey(reducer).mapValues(computemean).cache()
sig0mean_cache.saveAsTextFile('file:///home/skyfall/tmp/res.txt')
print sig0mean_cache.count()

# Reuse previous dataset in memory, applying data filter
print sig0mean.filter(filter_wind).count()
/tmp/coloc_spark.py                                   28,1          Haut
```

DATAVORE using HBase interactive requests

Metadata Indexation...

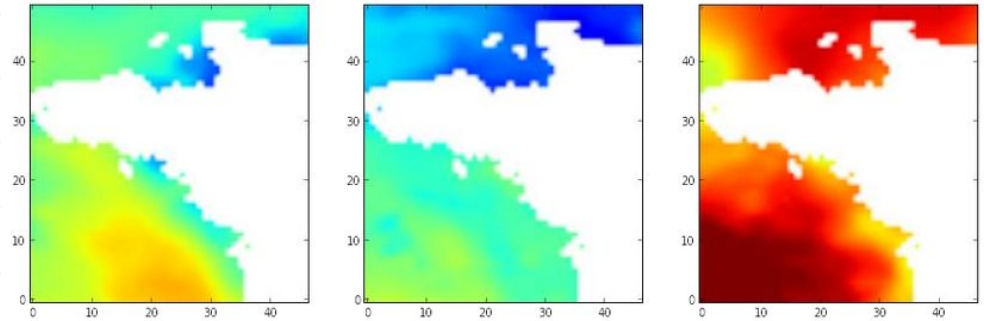
Q°: Precipitation over years, on this world region ?

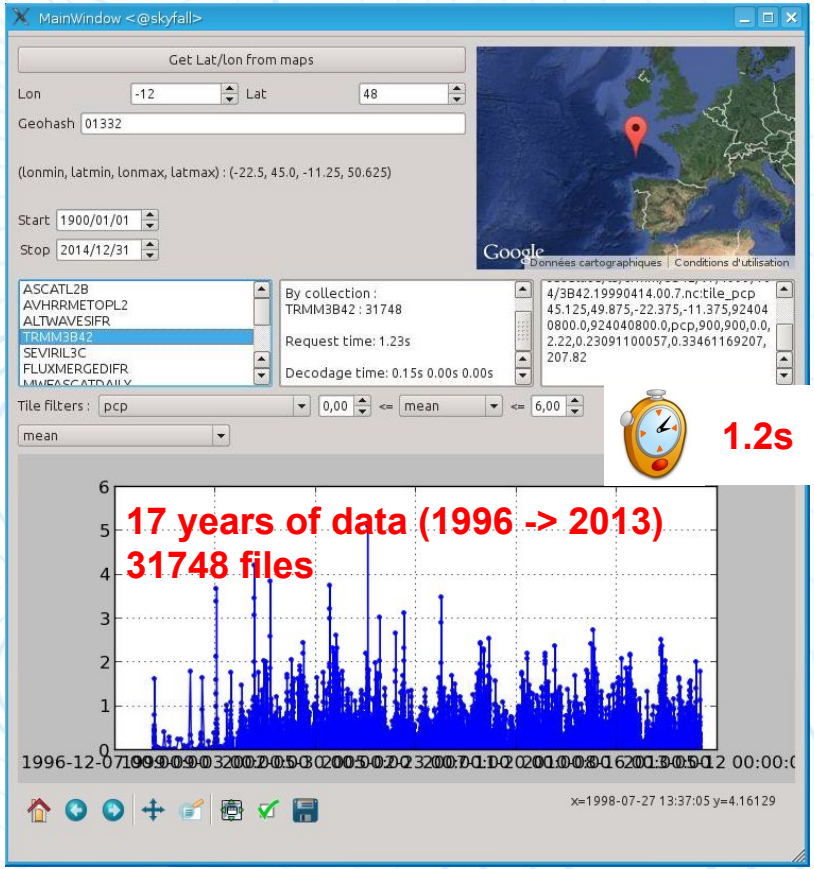
A° : < 2s to get long timeseries on tiles (5°x10°)

… but also Data indexation !

```
%time map = list(c.scan_data_Zone('ODYSSEAL4', 'analysed_sst', latmin = 45,lonmin = -4.81, \
                                  latmax=50, lonmax=-0.1, startdate='20070101', stopdate='20080101'))
print len(map) # 731
# affiche la carte sur la bretagne
subplot(1,3,1); imshow(map[0][1][2], origin='lower'); clim(9,17);
subplot(1,3,2); imshow(map[200][1][2], origin='lower'); clim(9,17);
subplot(1,3,3); imshow(map[600][1][2], origin='lower'); clim(9,17)
```

```
Wall time: 835 ms
731
```





QUESTIONS