PyTROLL

SMHI

Distributed EO satellite data processing with Pytroll/Satpy

...

Salomon Eliasson, Martin Raspaud, Adam Dybbroe
SMHI

martin.raspaud@smhi.se

PyTROLL

What is Pytroll/Satpy?

- Pytroll is a collection of free and open source python modules
- For reading, processing and writing EO satellite data
- Satpy is an easy to use front-end module, eg to generate imagery

Sentinel 2A, MSI

martin.raspaud@smhi.se

PyTROLL

Pytroll/Satpy in action

```
from glob import glob
from satpy.scene import Scene

# Load data by filenames
files = glob(“/data/himawari-8/*”)
scn = Scene(reader="ahi-hrit", filenames=files)
```

martin.raspaud@smhi.se

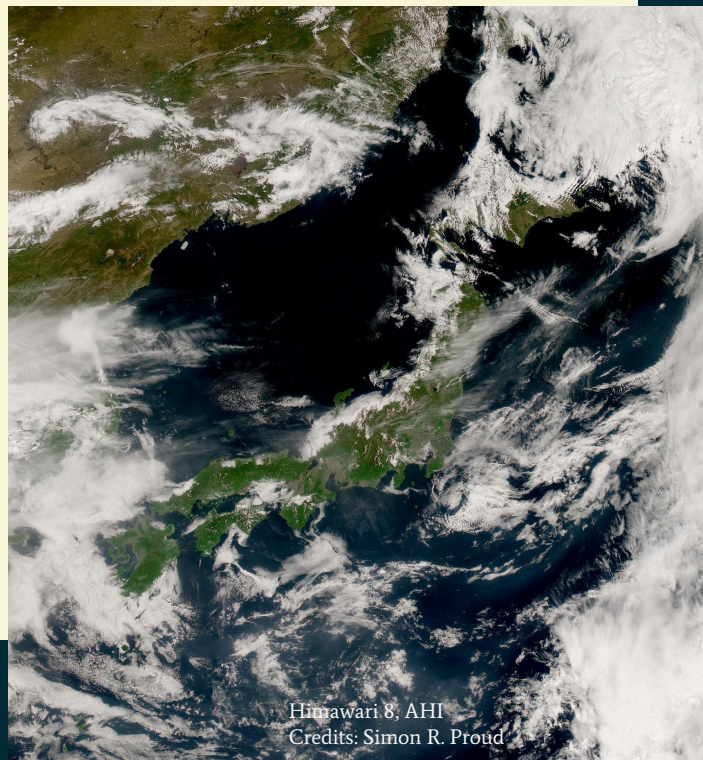
PyTROLL

Pytroll/Satpy in action

```
# Automatically load composites and their dependencies
scn.load(["true_color"])

# Resample multi-band data to a uniform grid
rs_scn = scn.resample("japan")

# Save RGB geotiff
rs_scn.save_dataset(“true_color”)
```



martin.raspaud@smhi.se

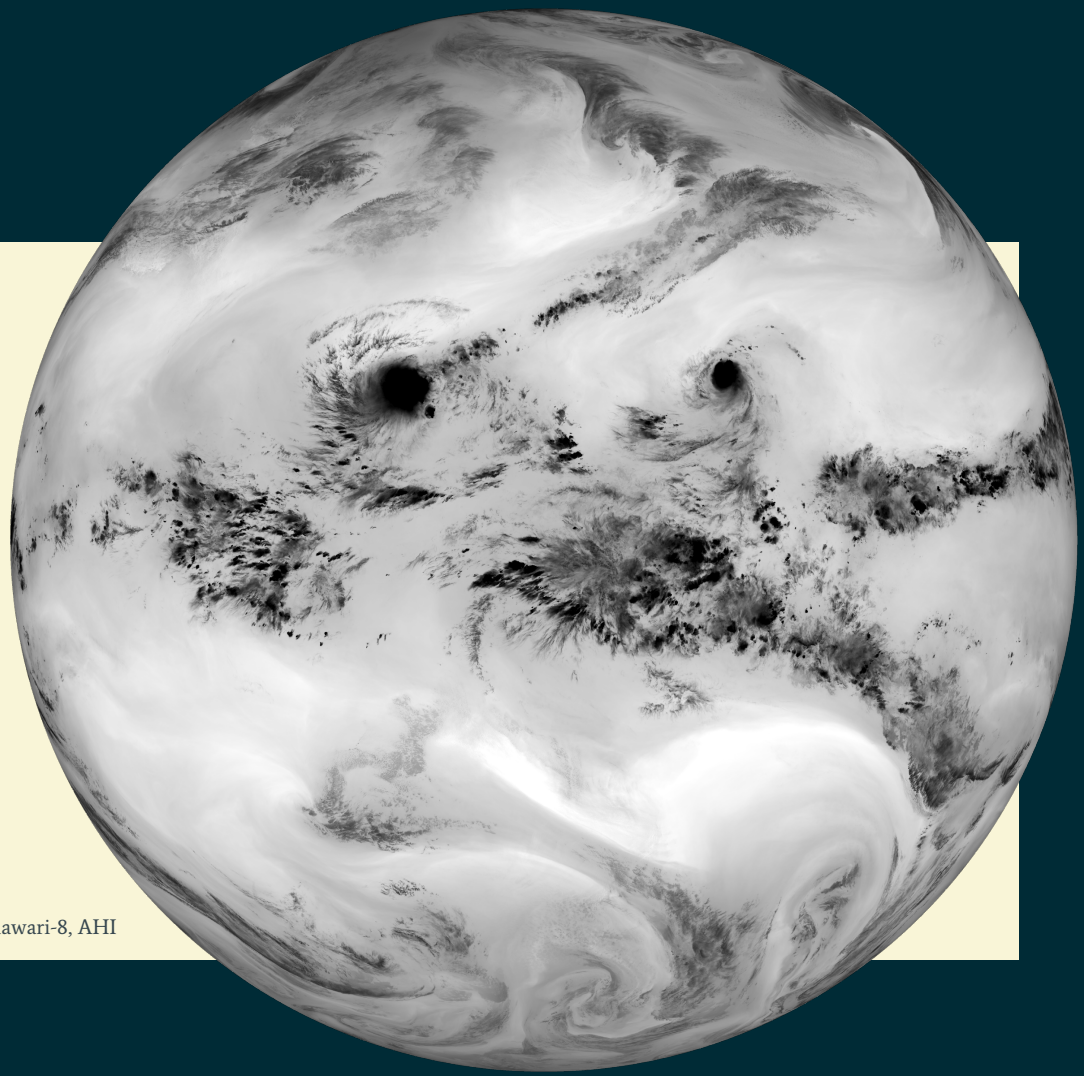
PyTROLL

Pytroll/Satpy in action

```
# Load single channels
scn.load(["B10", 0.6])

# Show a channel
scn.show("B10")

# Channel arithmetics
array = scn["B10"] + scn[0.6]
```



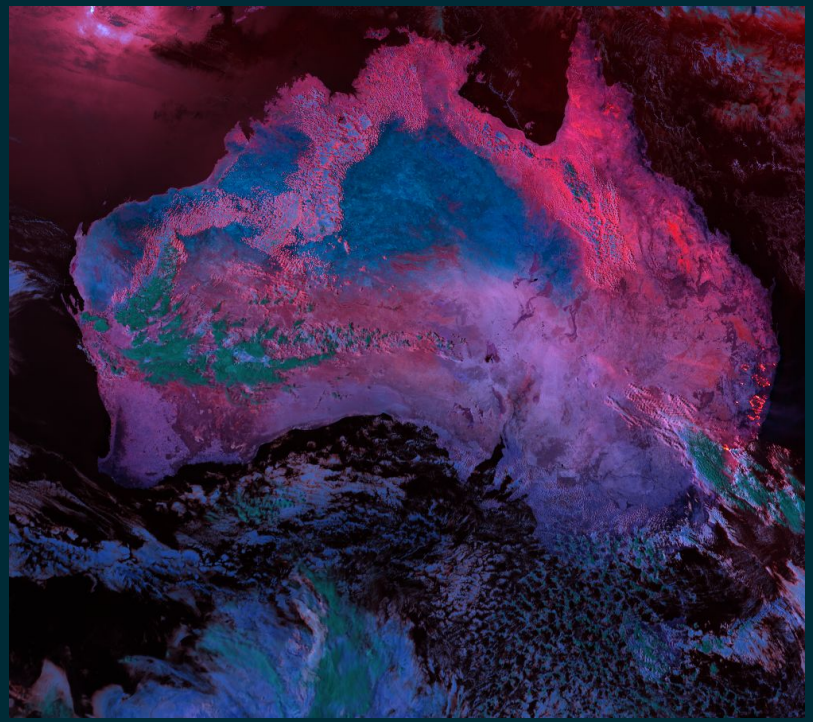
Himawari-8, AHI

martin.raspaud@smhi.se

PyTROLL

SatPy

- High level processing for satellite data
- Both GEO and LEO
- Indexing of channels by name or wavelength
- Many built-in composites
- Read many input formats
- Write many output formats
- Resample data to any PROJ.4 projection



Himawari-8, AHI, fire temperature product

martin.raspaud@smhi.se

PyTROLL

SMHI

The challenge

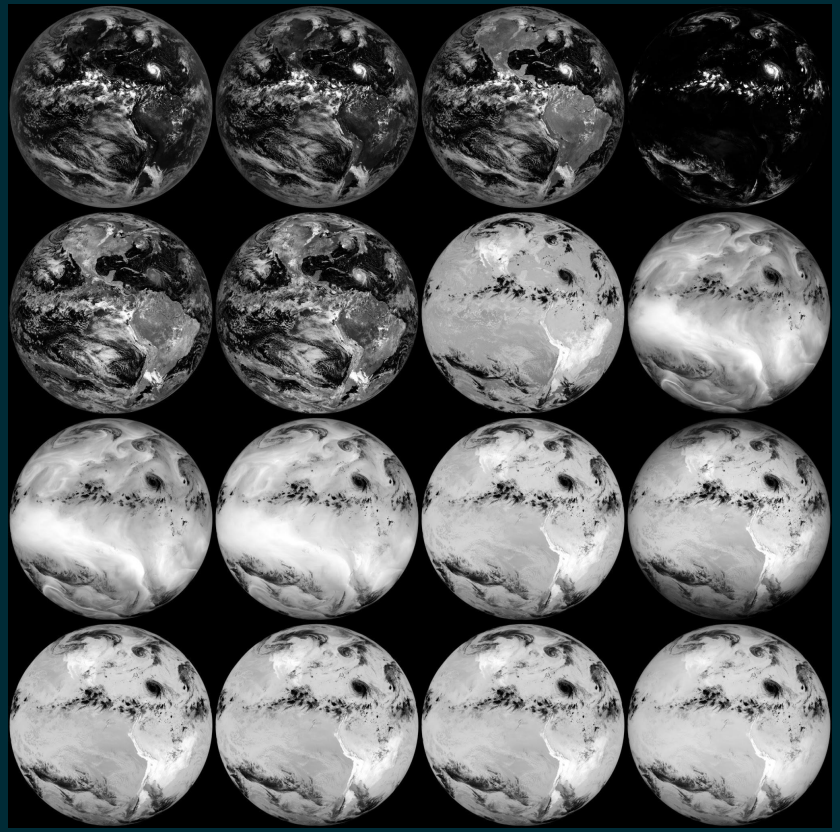
martin.raspaud@smhi.se

PyTROLL

SMHI

New Missions, Much more data

- GOES 16 and 17 ABI (3x, 4x, 5x)
- Himawari 8 and 9 AHI
- Sentinel 1, 2, 3
  in the order of 10 000 x 10 000
  pixels per segment
- EPS-SG
- MTG FCI



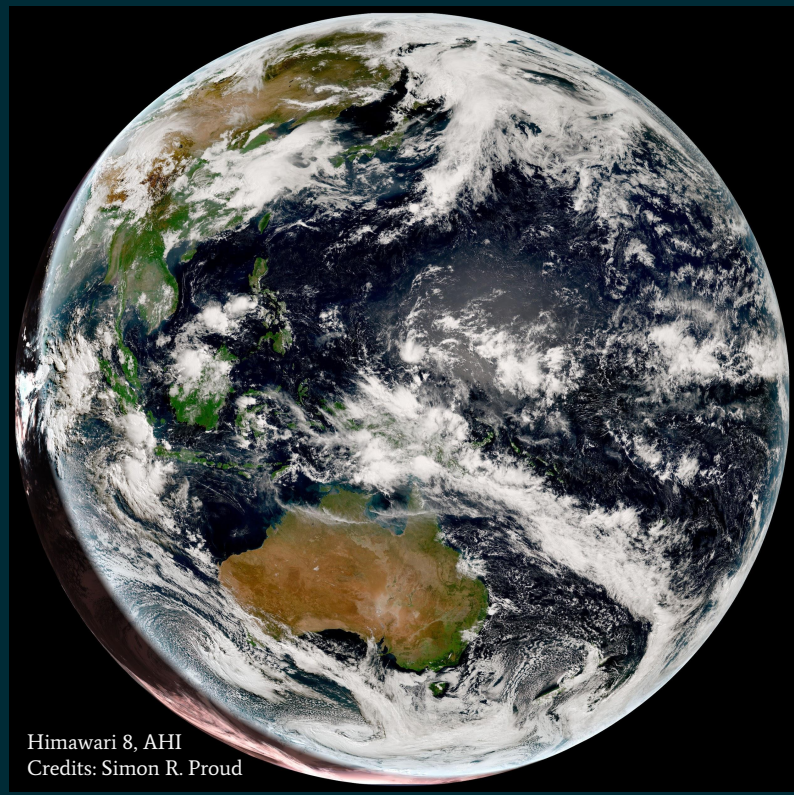
GOES-16, ABI

martin.raspaud@smhi.se

Data-size problem

- Too much data to fit in memory of regular computers
- Too long processing times due to single-threading

High Res!



Himawari 8, AHI
Credits: Simon R. Proud

PyTROLL

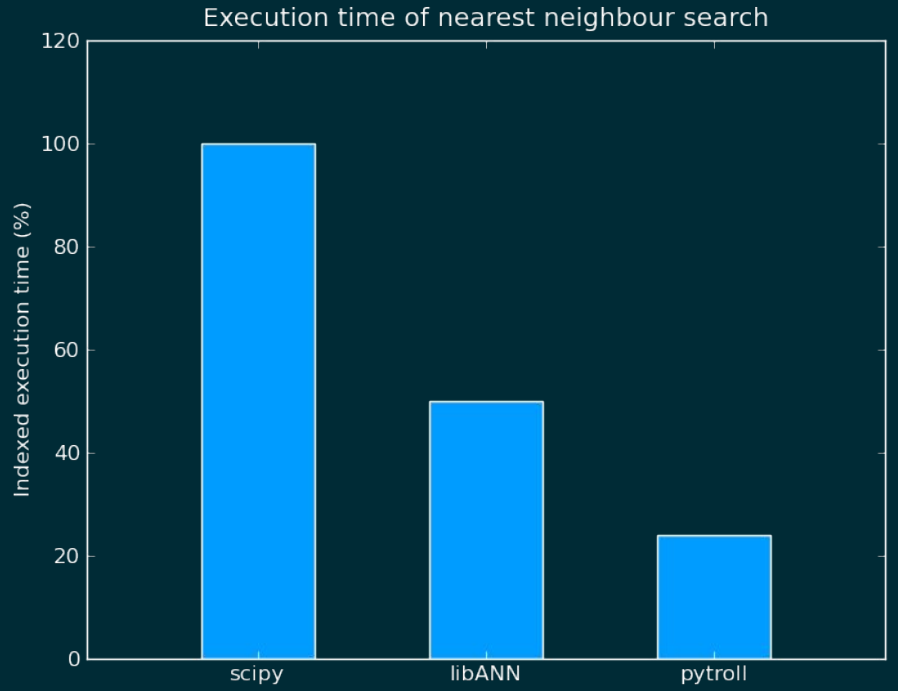
SMHI

Mitigation:
Optimized data processing

martin.raspaud@smhi.se

Efficient tools increase performance

- Python Scientific Stack:
  Numpy, Scipy
- Resampling:
  Pyresample & Pykdtree
- Tiepoint interpolation:
  Python-geotiepoints
- Spectral-domain computations:
  Pyspectral
- Orbital and space computations:
  Pyorbital



Pyresample resampling performance
vs Scipy and libANN

martin.raspaud@smhi.se

Mitigation:
Out of memory computations

martin.raspaud@smhi.se

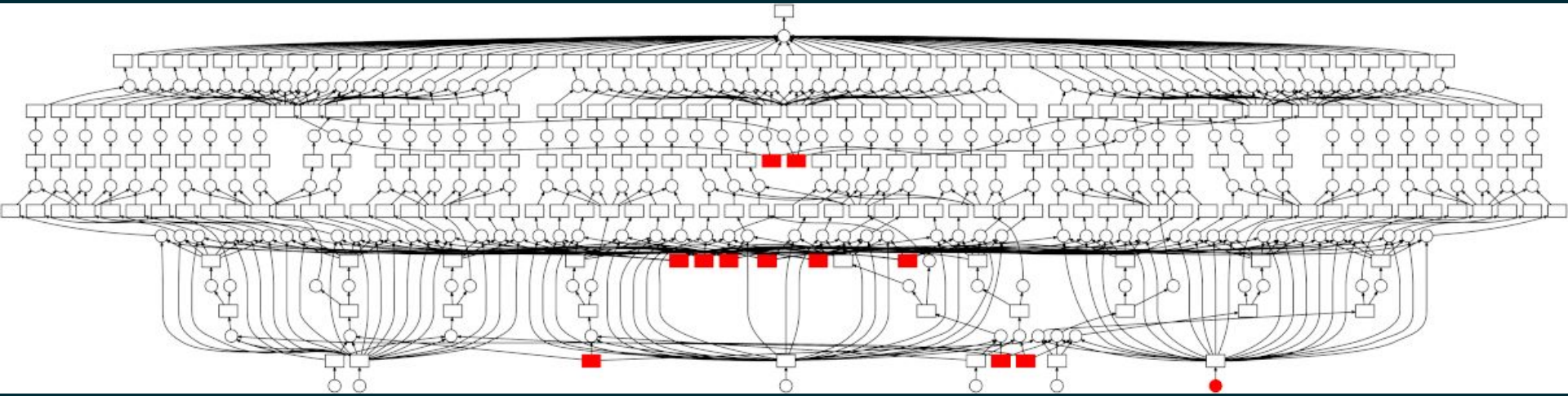
PyTROLL

SMHI

Using Dask for parallel computations

DASK

- Lazy processing
- Out-of-memory/Chunked processing
- Implements the numpy array interface



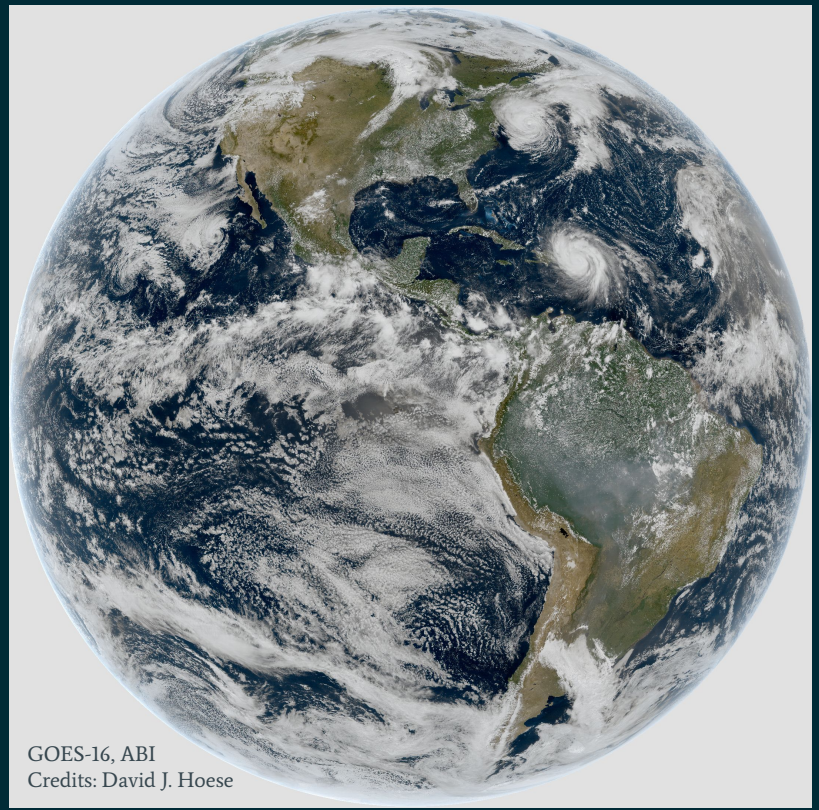
Example of chunked processing over time

martin.raspaud@smhi.se

Some performance results

- SatPy 0.8.4 - single core numpy
  - First execution crashed at 30m just before saving to geotiff
  - Total Time: ~23m (disk cache)
  - Peak Memory Usage: ~103GB
  - Time spent on I/O: 6m14s

- SatPy 0.9.0a1 - 8 Worker Threads (Dask):
  - Total Time: 5m38s
  - Peak Memory Usage: ~12GB
  - Time spent on I/O: 3m1s



GOES-16, ABI
Credits: David J. Hoese

martin.raspaud@smhi.se

PyTROLL

SMHI

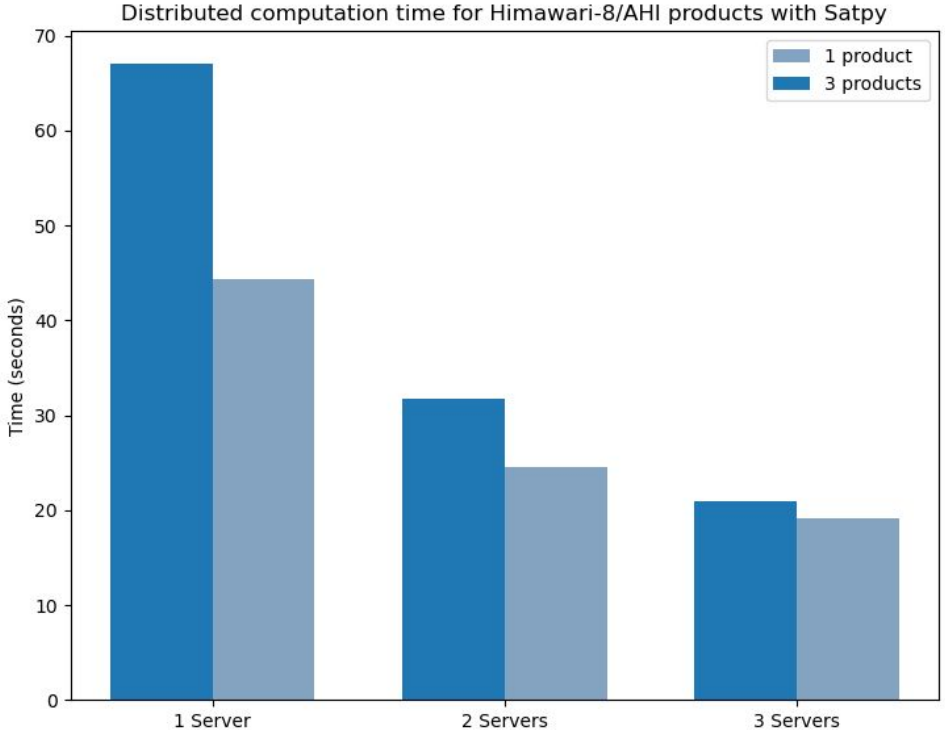
Mitigation:
Distributed processing

martin.raspaud@smhi.se

Dask distributed

DASK

- Client/Server architecture
- Works automatically on regular dask code
- Works on clusters



Sentinel 2A, MSI

The Pytroll Philosophy

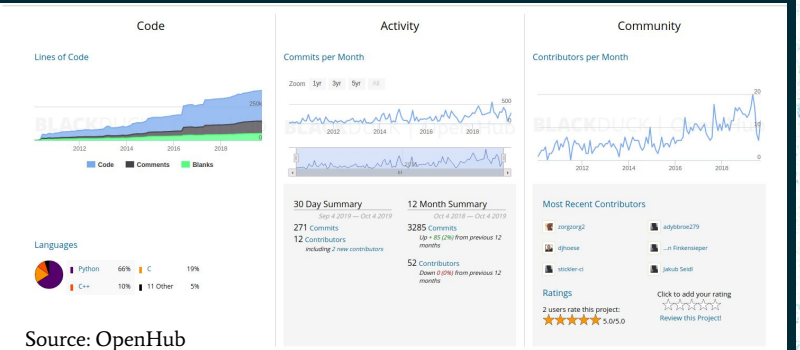
martin.raspaud@smhi.se

PyTROLL

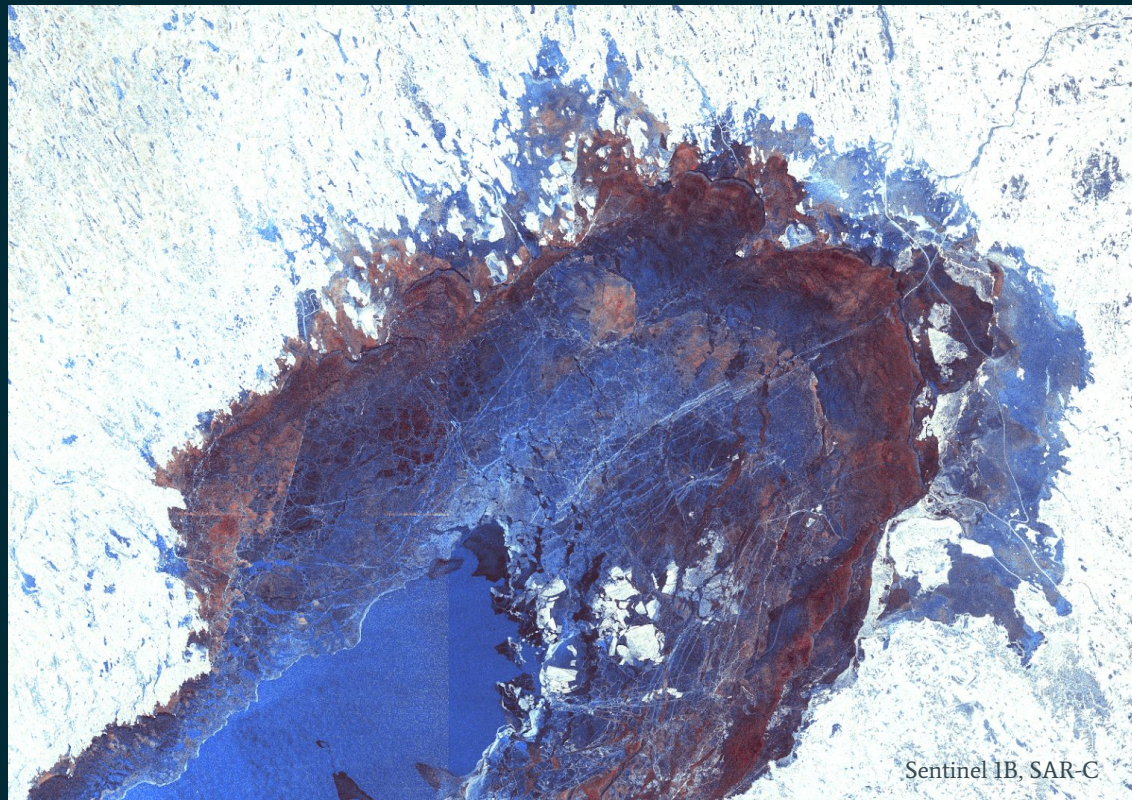
SMHI

Pytroll

- FOSS ((L)GPL, Github)
- Agile development (CI, Code reviews)
- Active community (> 100 contributors, Hackathons)



Source: OpenHub



Sentinel 1B, SAR-C

www.pytroll.org

Pytroll@Slack

Pytroll@Gitub

pytroll@googlegroups.com

PytrollOrg@Twitter

Thanks !

Sentinel 2B, MSI