Reading and processing VIIRS data efficiently with Pytroll

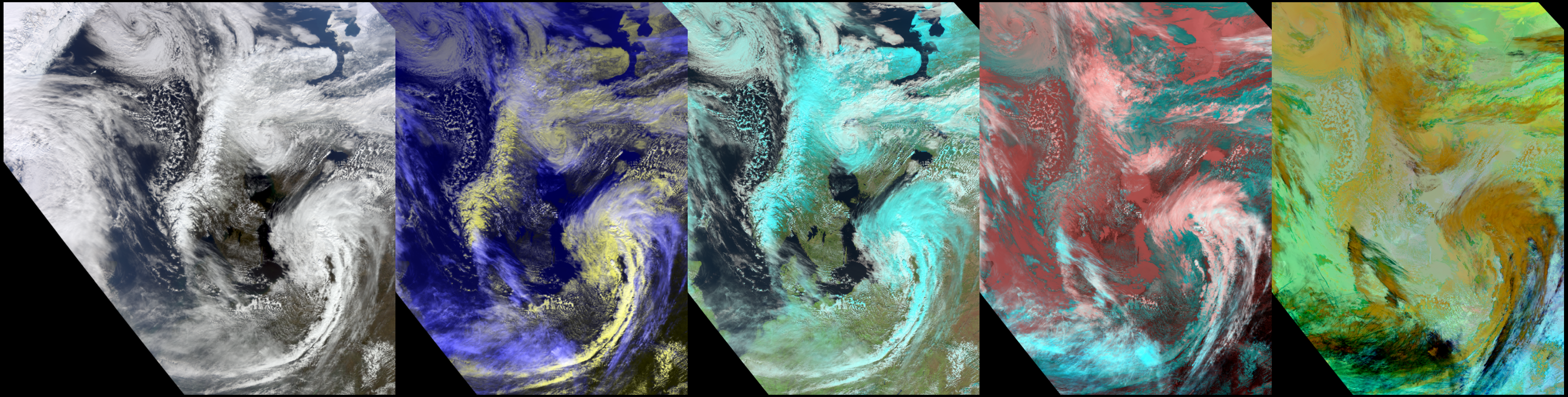
Martin Raspaud, Adam Dybbroe (SMHI)

Pytroll summary

- Community driven FOSS python packages
- Satellite data reading & processing
- Users include DMI, FMI, IMO, SMHI, CNMCA, DWD, KNMI, MeteoSwiss, ZAMG, Met.no, EC...

Objectives

- Batch processing
- Both for operations and research



martin.raspaud@smhi.se

VIIRS preprocessing (-> SDR)

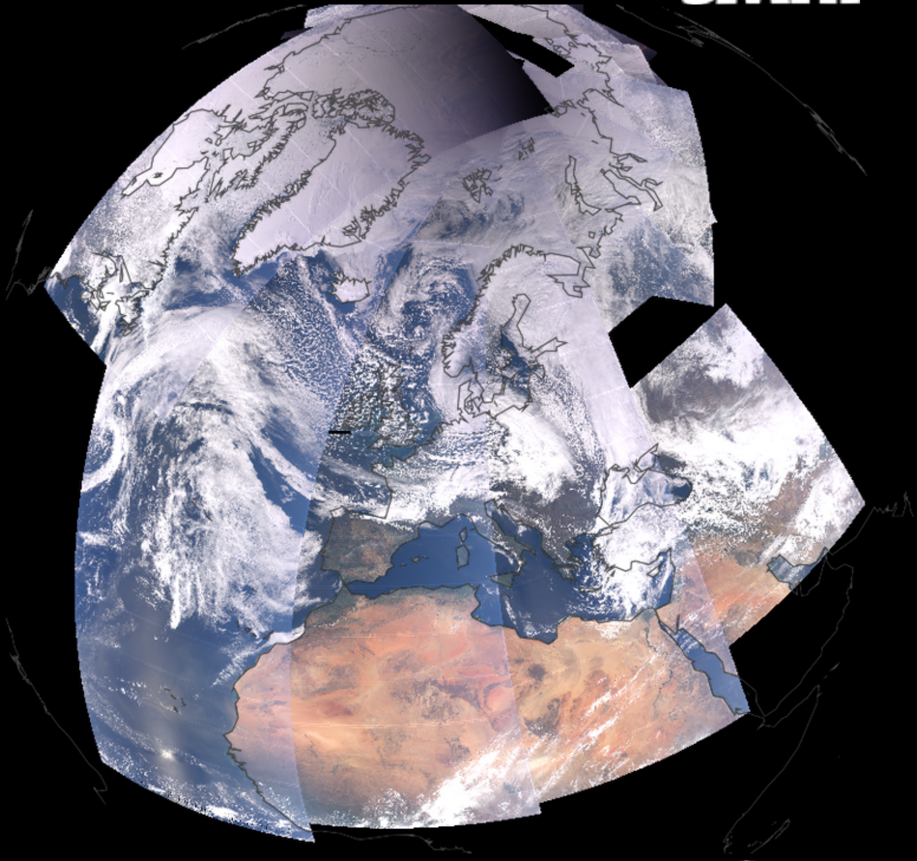
- CADU streaming into RT-STPS
- RDR granules processed sequentially…
- … by CSPP

Can’t be much faster !

Reading in mpop

- SDRs: viirs-sdr reader
- Compact SDRs: viirs-compact reader

Which granules ?

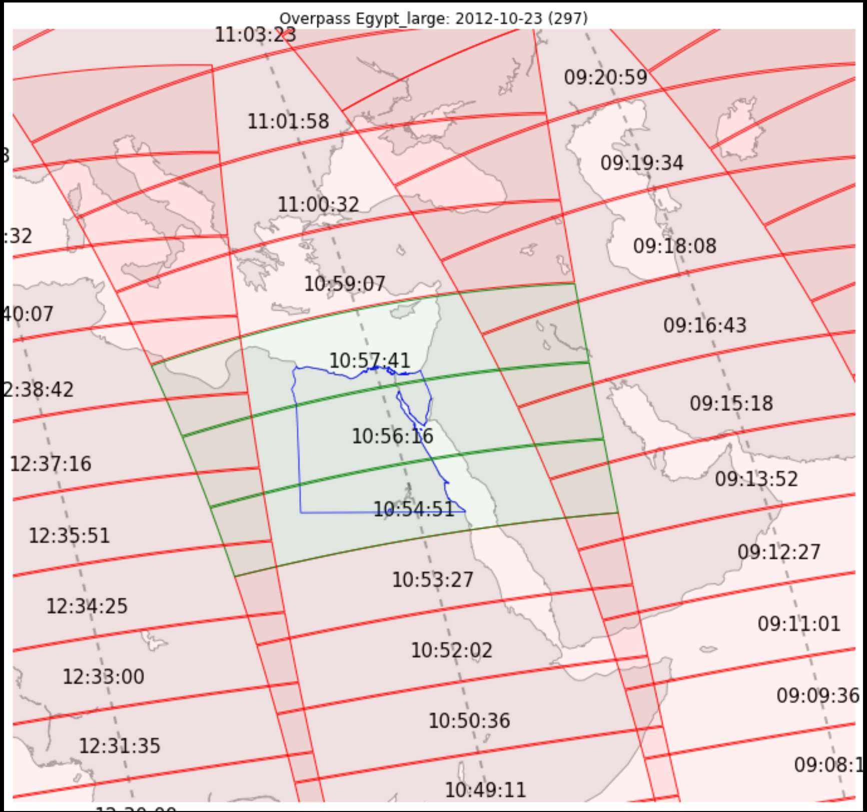


martin.raspaud@smhi.se

Using pyresample

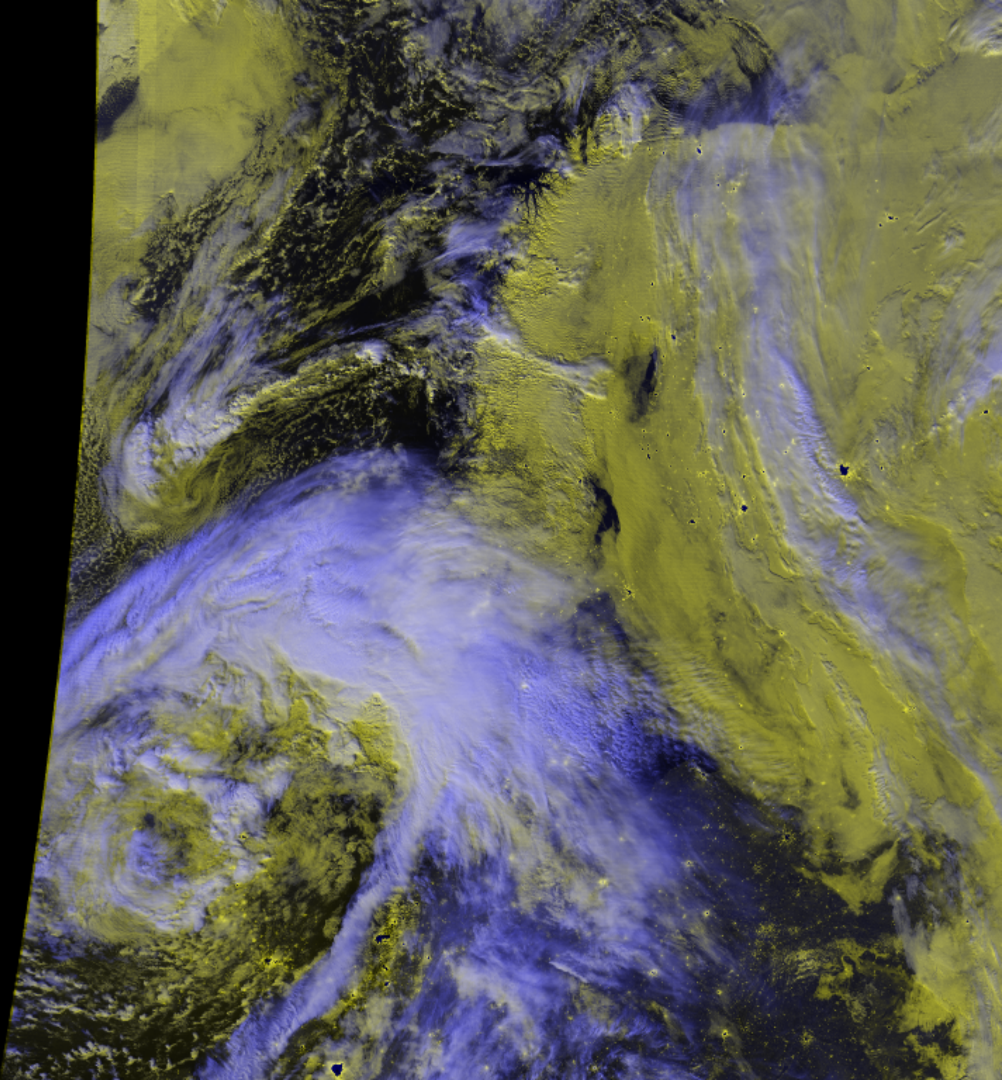
- Load only granules within area of interest

```
scene.load([“M01”, “M02”,
“M05”], area=”europe”)
```



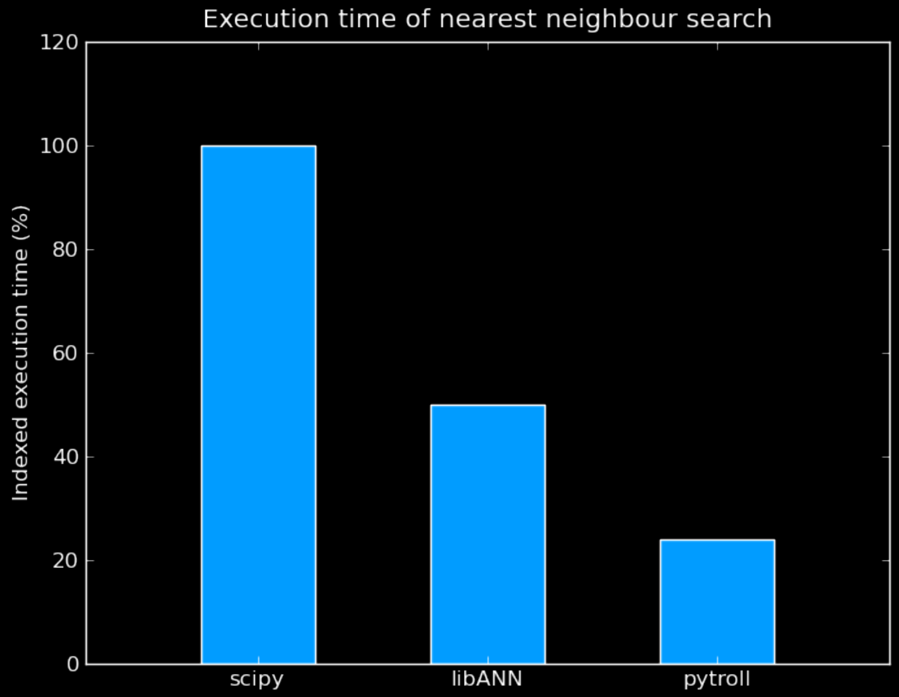
Fast loading

- Optimization of memory usage
- no more than 50 MB overhead
- Geolocation is read only once



Pyresample optimization

- Nearest-neighbour is fast and generic…



martin.raspaud@smhi.se

Pyresample optimization

- ...but not always the most efficient
- Use gradient search instead [1]

[1] Implementation and Evaluation of Concurrent Gradient Search Method for Reprojection of MODIS Level 1B Imagery, Khlopenkov & Trishchenko, 2008

Pyresample optimization

- Written in Cython
- Up to 7× faster for difficult cases.
- I-band to 250m baltic sea (3 granules)
  - NN: 51.3 seconds
  - Gradient search: 6.5 seconds

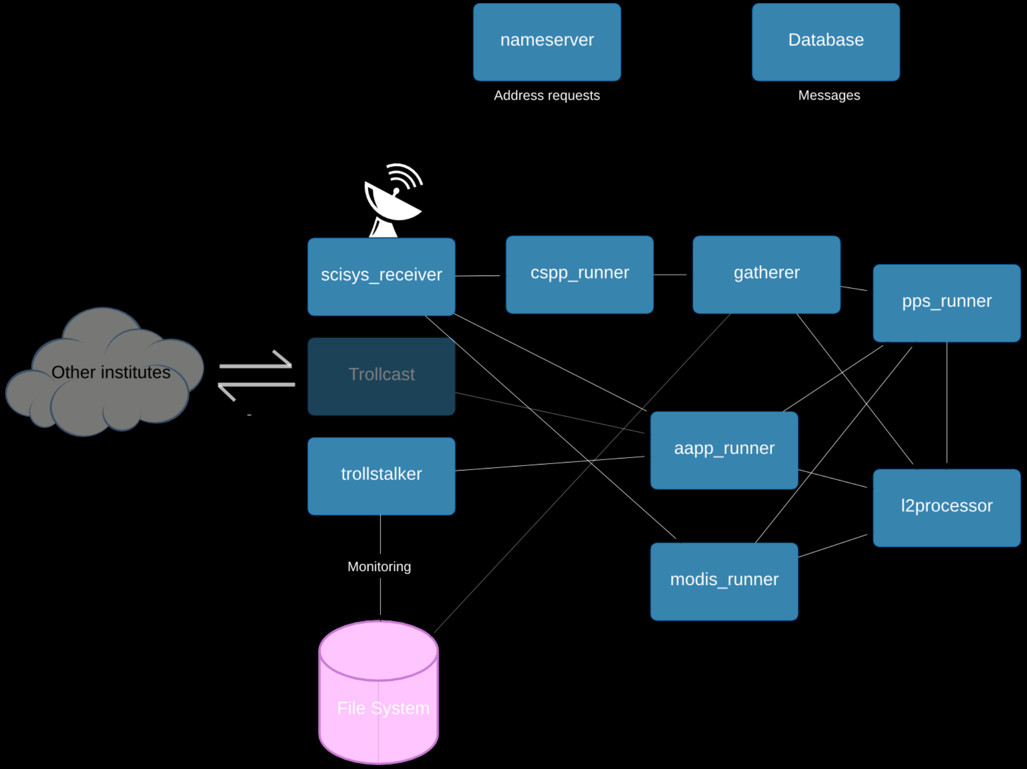
Pycoast optimization

- Pycoast loads data from GSHHS
- Caching for faster reuse

Batch processing

- New trollduction module
- Event-based (messaging)
- Product lists

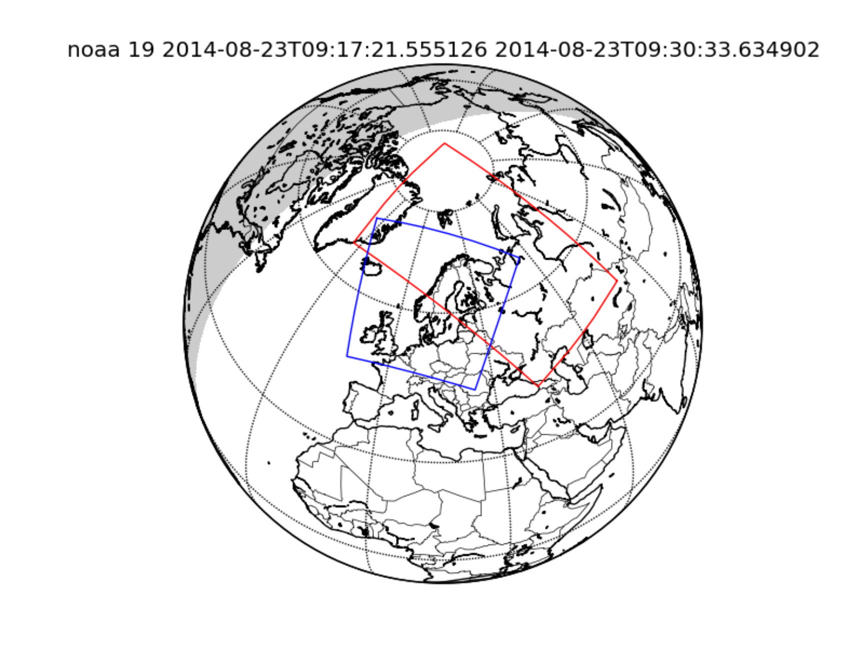
Batch processing



martin.raspaud@smhi.se

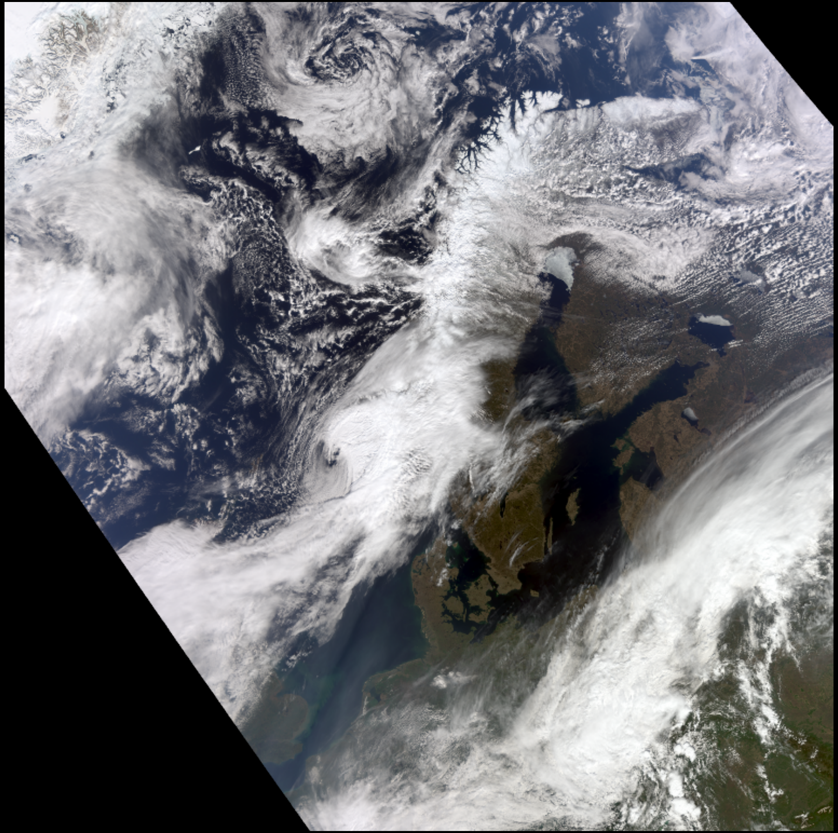
Overpass scheduling

- Based on area of interest
- Time complementarity
- Multi-antenna support
- WIP: Multi-station


noaa 19 2014-08-23T09:17:21.555126 2014-08-23T09:30:33.634902

martin.raspaud@smhi.se

Thank you for your attention!



Visit pytroll.org!

Any questions ?

martin.raspaud@smhi.se