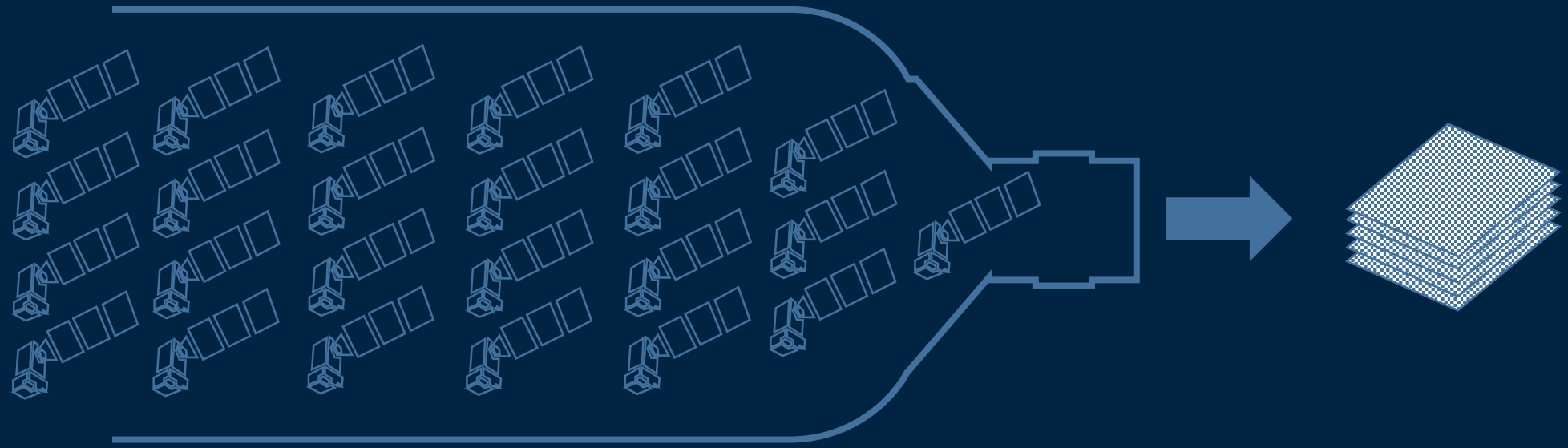
Local and automated processing of Sentinel-2 time series:



Adressing the bottlenecks

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF SCIENCES

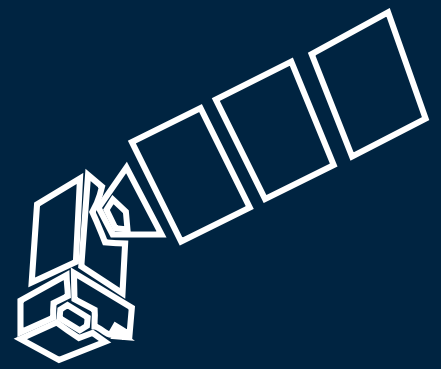
Philipp Hochreuther

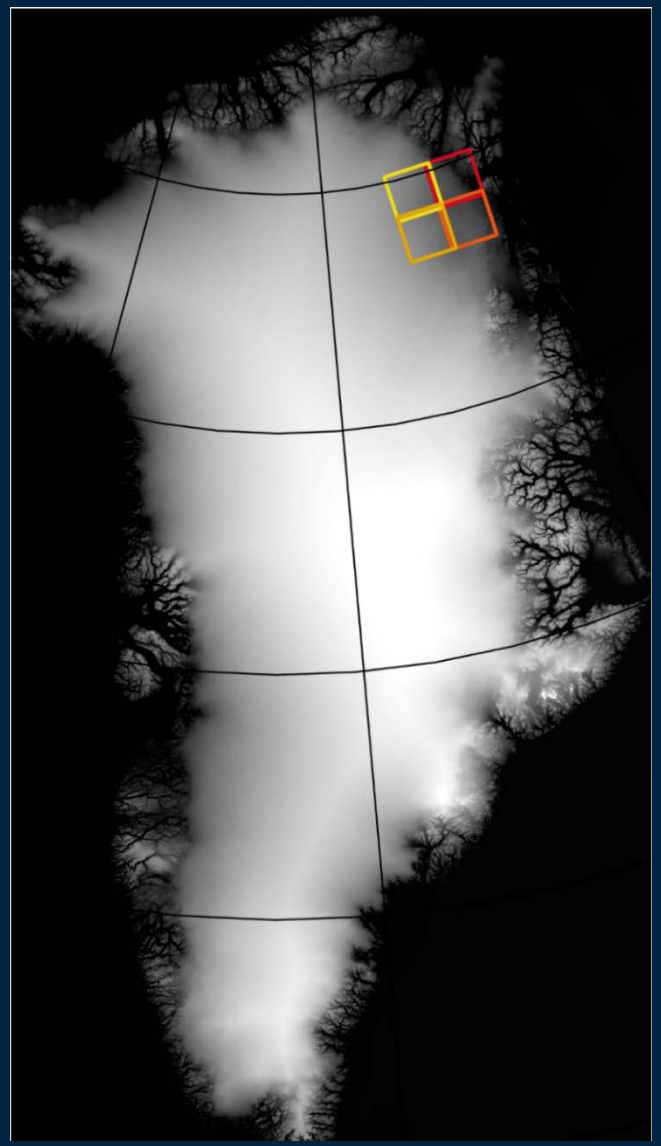
big data from space

21.03.2019

Overview

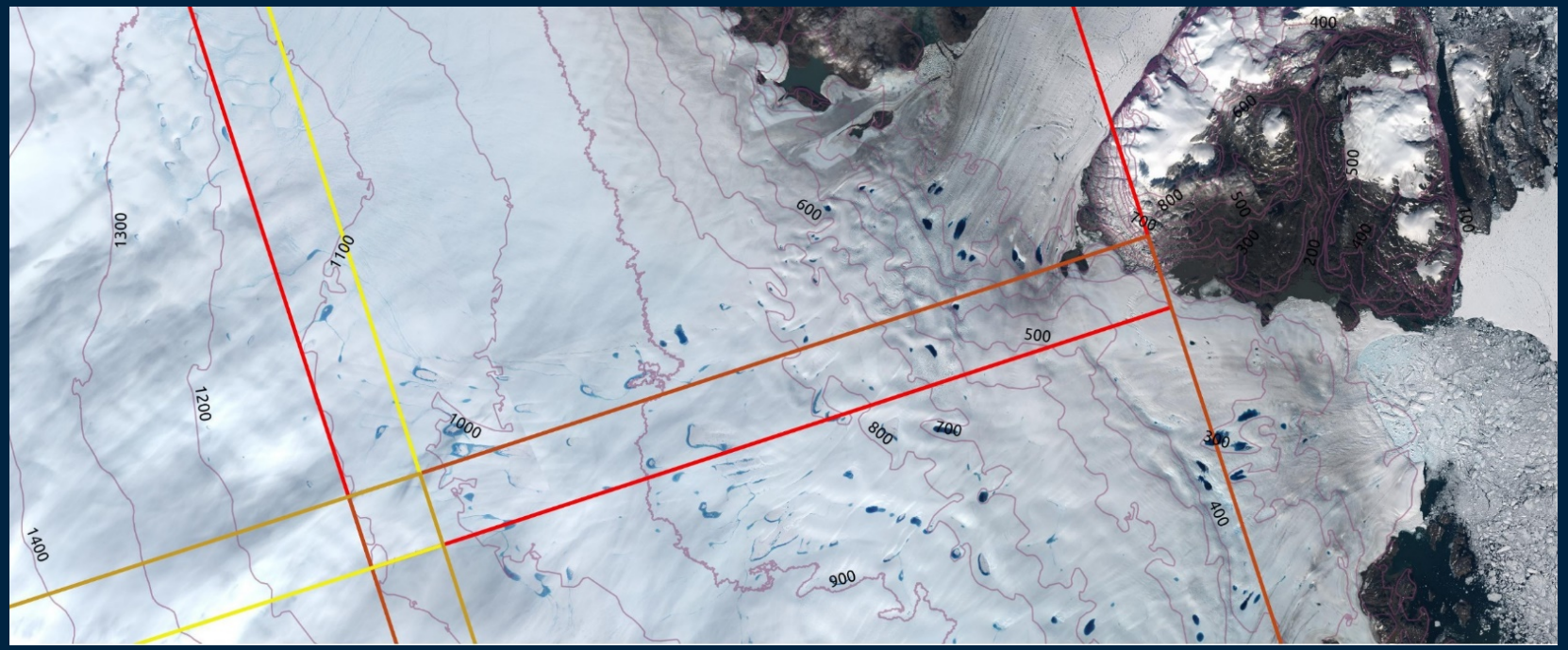
- Motivation and aims
- Hardware & setup
- Data
- Processing chain
- Bottlenecks
- Recommendations

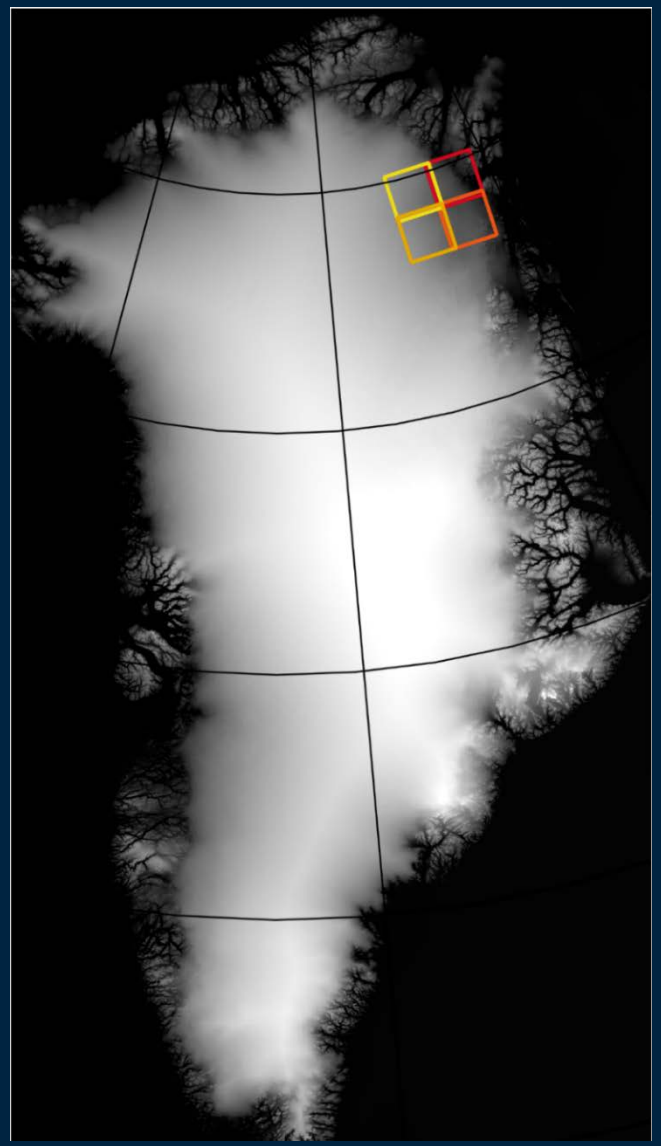




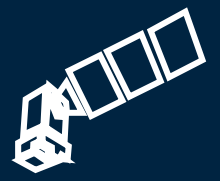
Scientific background:

- Glacier surface of 79°N glacier
- Multi-band analysis of optical satellite data (Sentinel-2 A/B)
- Time series analysis to detect intra- and inter-annual changes
- Mosaics of four S-2 granules necessary to cover the whole area of interest

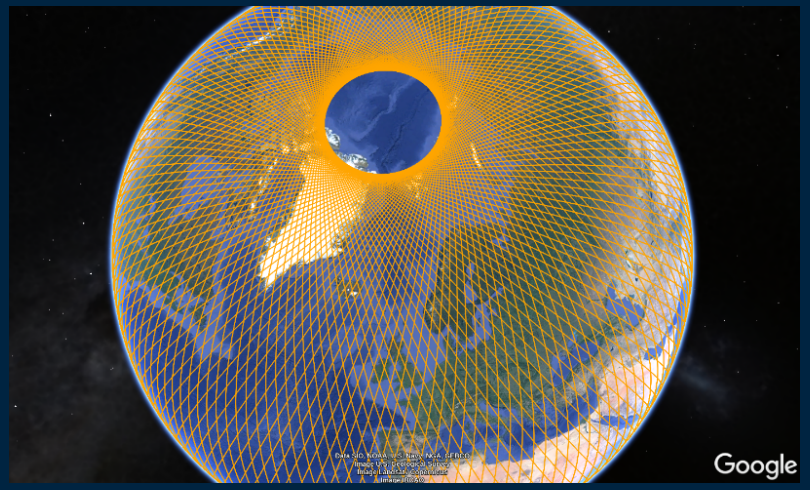


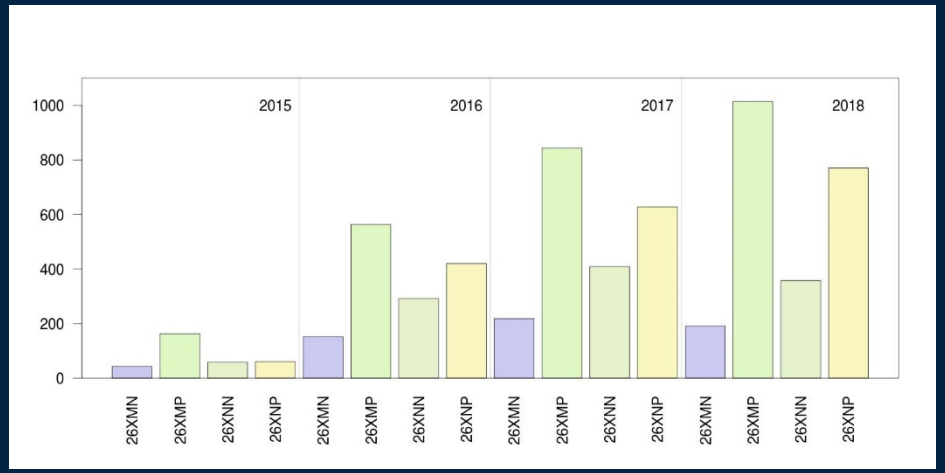


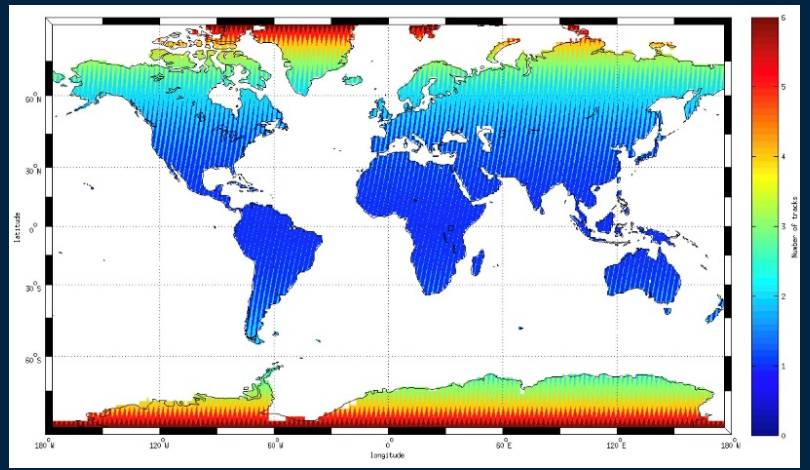
Sentinel-2 A/B



- S-2 A: June 23, 2015 | S-2 B: March 07, 2017
- Sun-synchronous orbits
- Global revisiting time: 5 days; revisiting time at 79°N: 1,39 days
- L1C: 600 MB/scene (ESA)

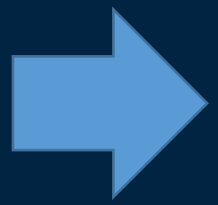






Sentinel-2 A/B

- S-2 A: June 23, 2015 | S-2 B: March 07, 2017
- Sun-synchronous orbits
- Global revisiting time: 5 days; revisiting time at 79°N: 1,39 days
- L1C: 600 MB/scene (ESA)



BIG

(a huge amount of)

DATA!

Obvious solutions:

amazon web services

Google Earth Engine

Microsoft Azure

Microsoft Azure

Great!

- Largely scalable
- Secure
- Huge spectrum of capabilities (of which most of them you‘ll never use)

… but what if:

- You have a well-defined application
- You need to use ressources over longer time without continuous income
- You are sceptical towards a centralization of the internet
- You (need to) care about data privacy

Aims:

- Develop a processing chain that:
  - Uses local ressources
  - Is based on free & open source software alone
  - Is scalable based on initial knowledge of AOI size
  - Is fully automated
- Identify bottlenecks in the processing chain and possible solutions
- Give hardware recommendations based on empirical knowledge

Colocation

Processing

2 CPUs à 12 cores
= 48 threads

126 GB RAM
1.2 TB SSD

Storage

1 CPU
126 GB RAM
165 TB HDD

ubuntu

GDAL

R

esa
Sen2Cor Processor

Processing

2 CPUs à 12 cores
= 48 threads

126 GB RAM
1.2 TB SSD

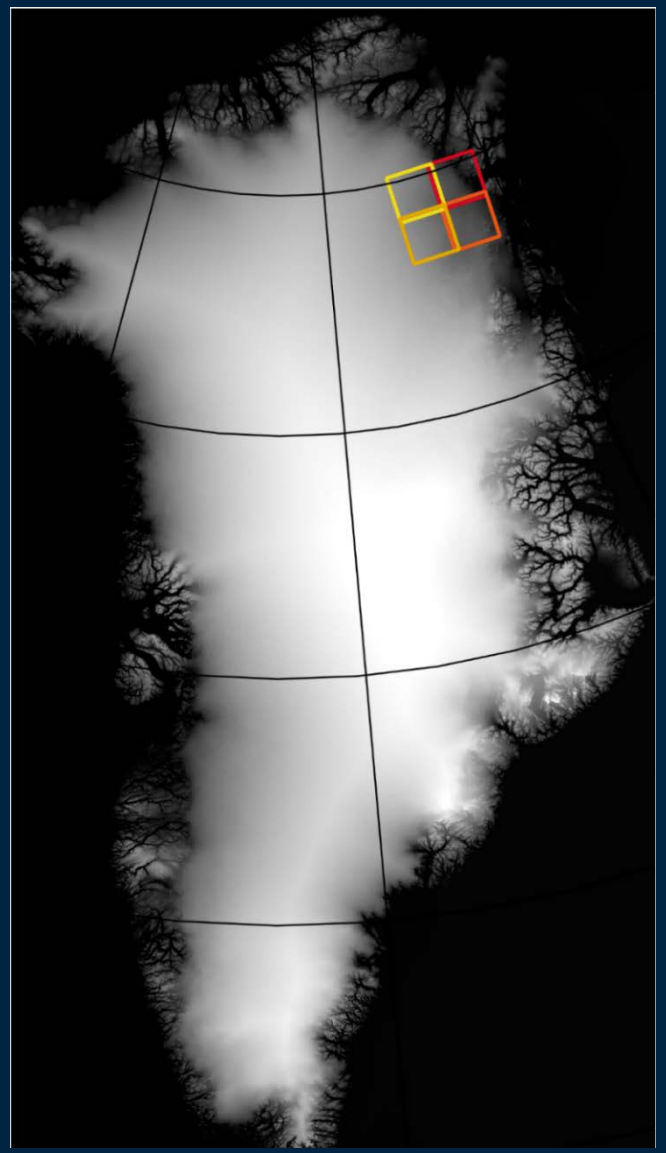
Storage

1 CPU
126 GB RAM
165 TB HDD

ubuntu

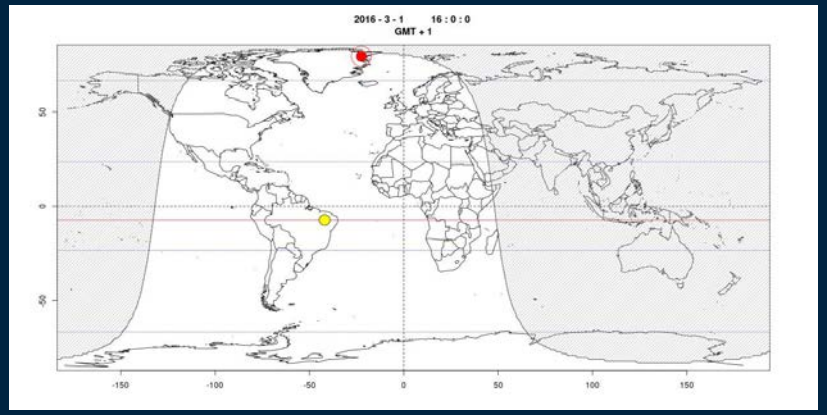
SAMBA

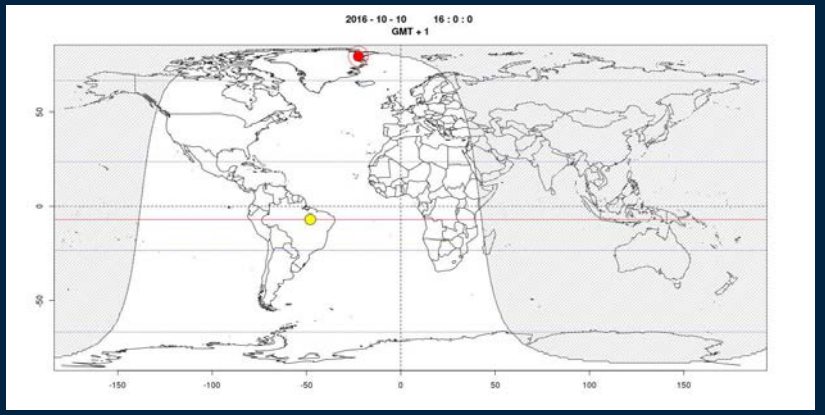
MUNIN



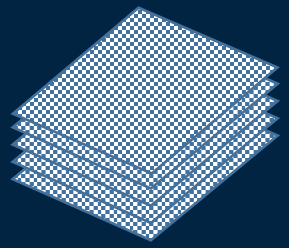
Sentinel-2 A/B

- Global revisiting time: 5 days; revisiting time at 79°N: 1,39 days
- L1C: 600 MB/scene (ESA)
- Granules: 26XMN, 26XMP, 26XNN, 26XNP
- Availibility of daylight: beginning of March to end of September



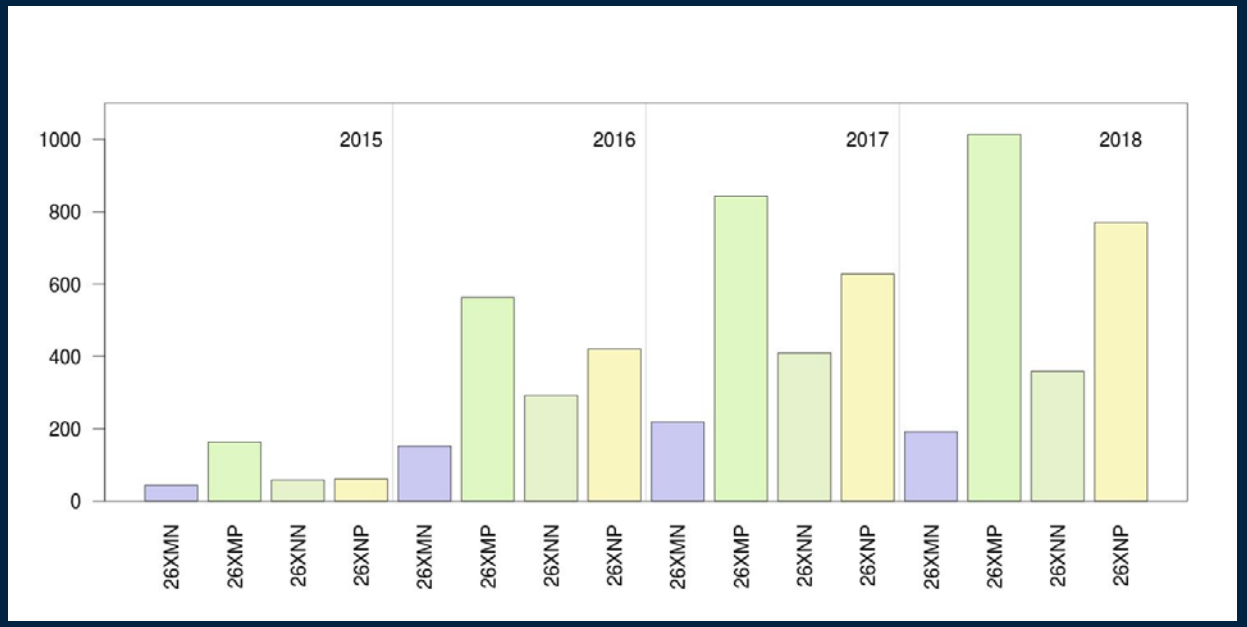


Data: some numbers…



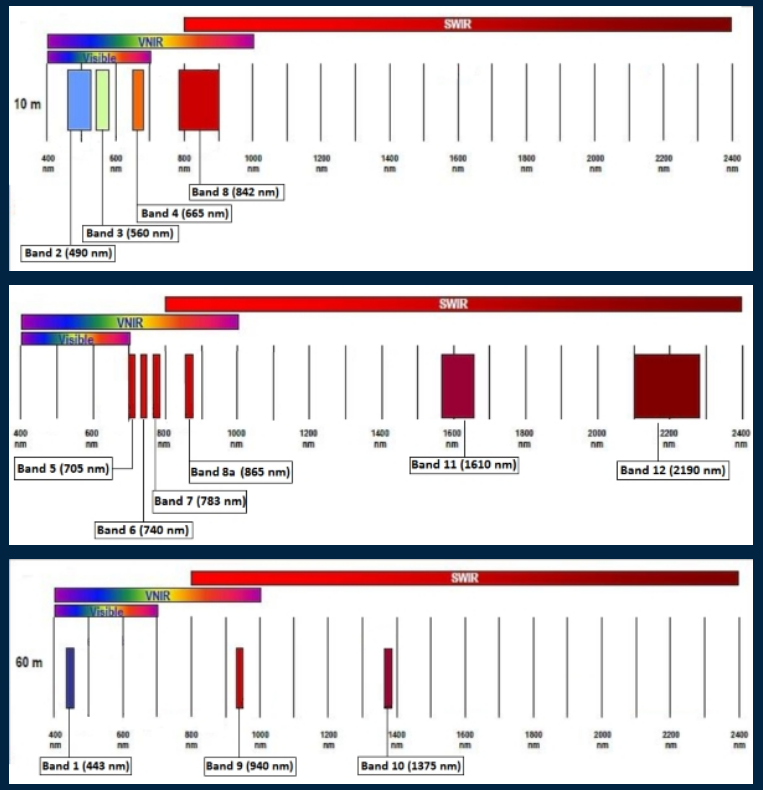
4 granules * $\frac{days(March\ to\ September)}{0.72\ scenes/day}$ * 3 years = 5.817 scenes

2.82 TB used space / 5.817 scenes = 508.64 MB/scene



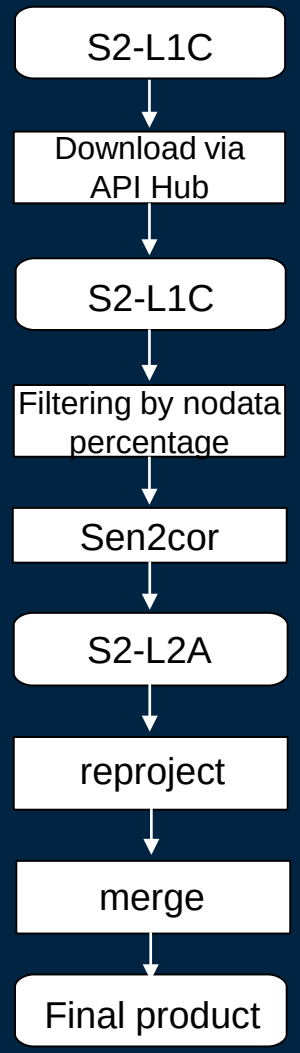
Desired end products:

- All bands available as L2A product, including QA bands
- Whole area of interest merged to one image per band

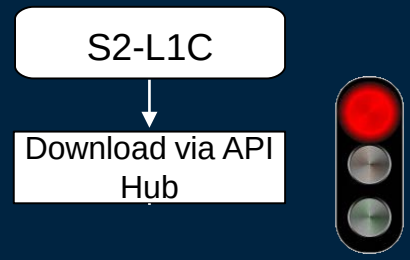


Test dataset: 4 granules

- Bands 2,3,4,8 @ 10m
- Bands 2,3,4,5,6,7,8A,11,12 @ 20m
- Bands 1,2,3,4,5,6,7,8A,9,11,12 @ 60m
- plus AOT and WVP at all resolutions
- 200 km x 200 km images



Data download



Script (Sentinel-download, available via Github)

HTTP request (1 per file)

Authentification

Download to HDD

Open API Hub

Total download time for 2.82 TB:
1094.75 hours
= 45.6 days

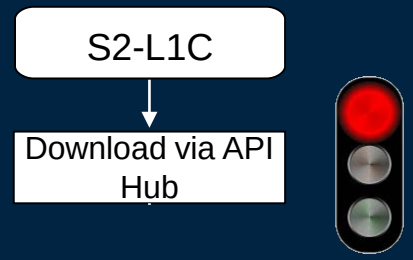
Data download

Alternative: Google Cloud SDK

One time (Ubuntu/Debian):

- Add Google Cloud SDK as software repository
- Apt-get install google-cloud-sdk

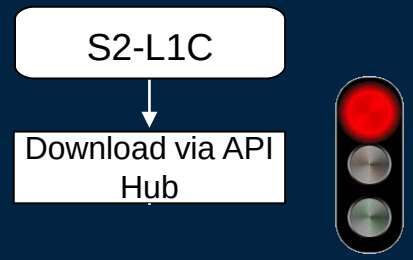
Download data with a one-liner (gc-copy)



Total downlaod time for 2.82 TB:
56.65 hours
= 2.36 days

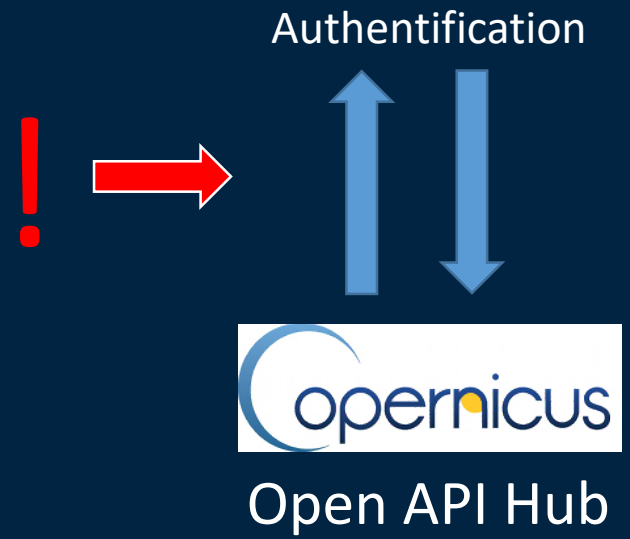
HTTP request (1 for all) → Cloud SDK → Download to HDD

Data download



1094.75 vs. 56.65 hours – are the Copernicus servers slow?

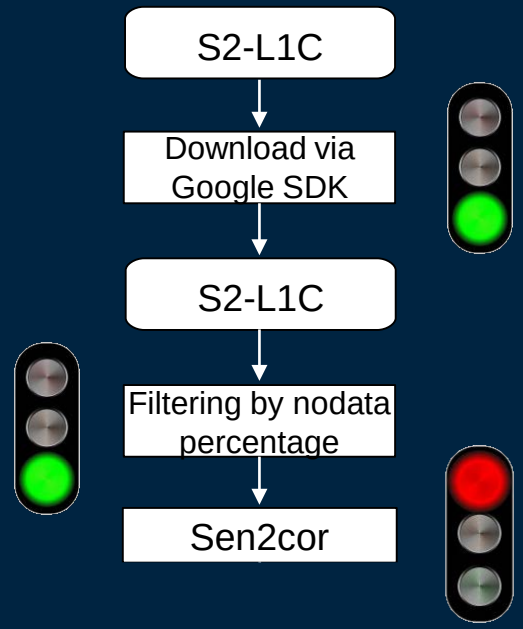
The answer is: NO!



Download speed Copernicus: 15 MB/s
Download speed Google SDK: 14.5 MB/s

Atmospheric correction – Sen2Cor

- Performs atmospheric-, terrain- and cirrus correction
- L1C to L2A
- Quality indicators, classification maps
- Available as standalone or included in SNAP, with or without GUI



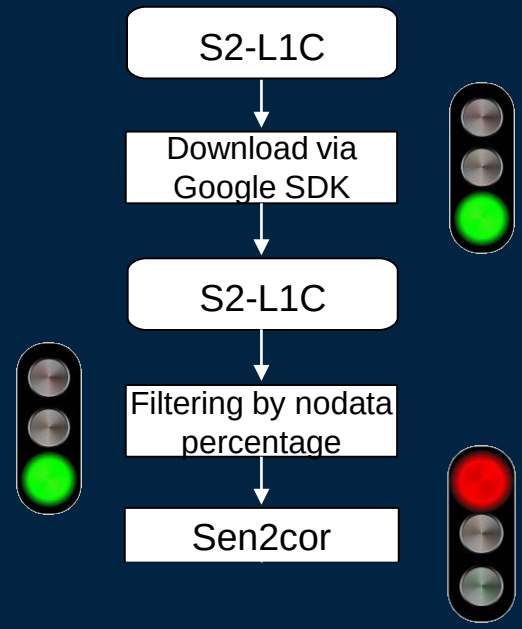
Sequential run: 15 – 30 minutes, 1 thread, max. 4 GB main memory usage

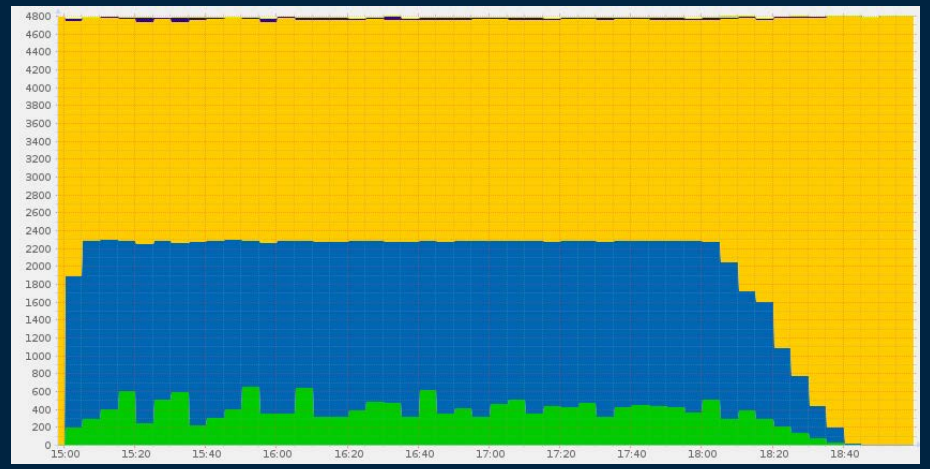
→ 484.6 hours/20.2 days for 5.817 scenes

Atmospheric correction – Sen2Cor

- Solution: parallelization

- Available threads: 48 → max. 47 parallel processes
- Available RAM: 128 GB → max. 31 parallel processes

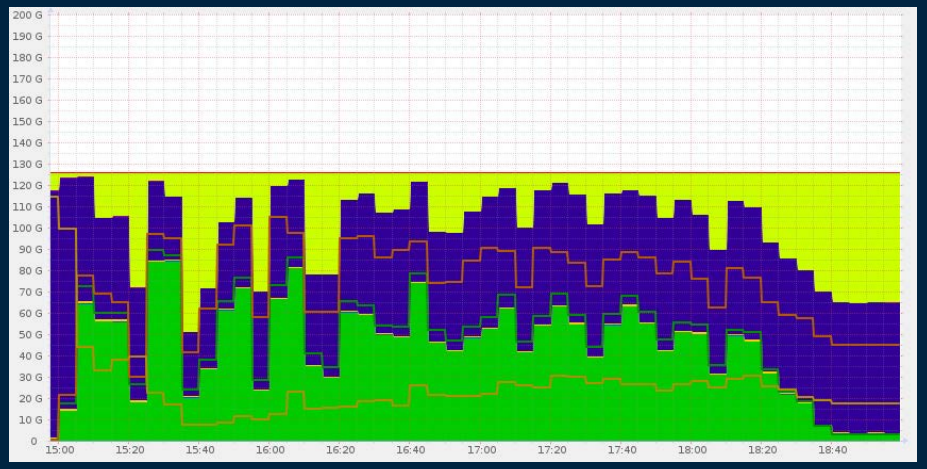


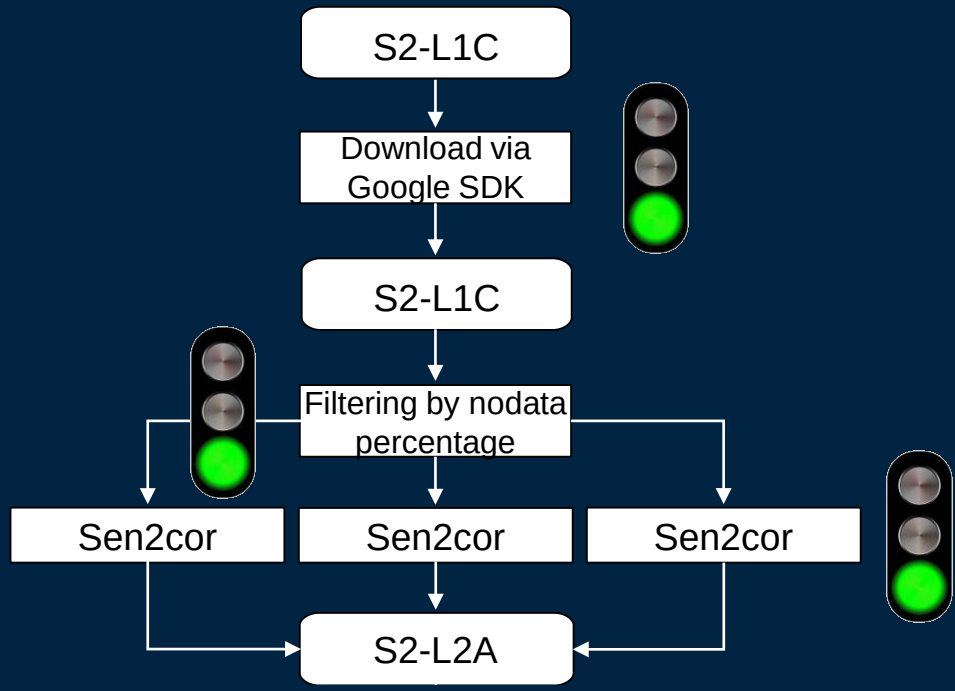


Atmospheric correction – Sen2Cor

- Solution: parallelization

- Maximum RAM usage with 24 cores: 85 GB
  → 3.54 GB/process
- Total runtime for 1117 scenes: 3h 40 min
  → 5.1 min/scene





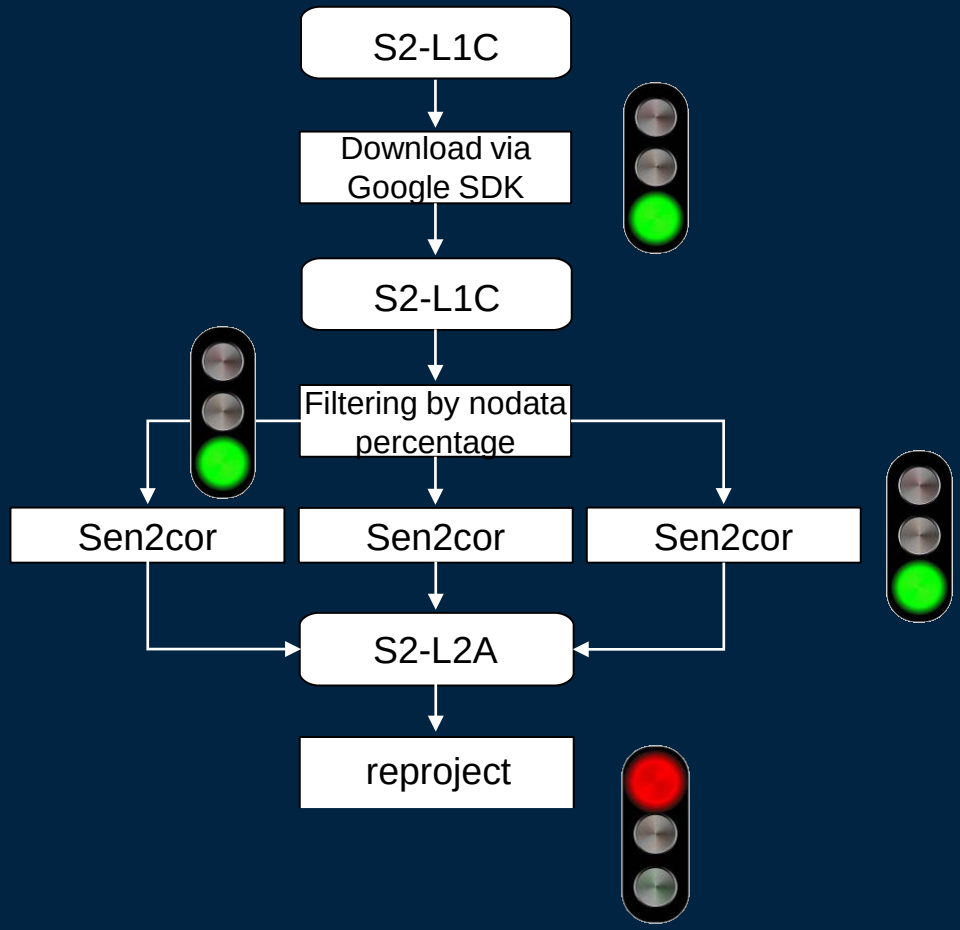
Reprojection

- S-2 data comes in UTM (zone 26)
- Reprojection to NSIDC polar stereographic north (EPSG 3414)
- Sequential run (gdal_warp): 160 sec/scene

→ 5.817 scenes: 10.78 days

Solution (again): Parallelization

→ less than 24 hours for this step

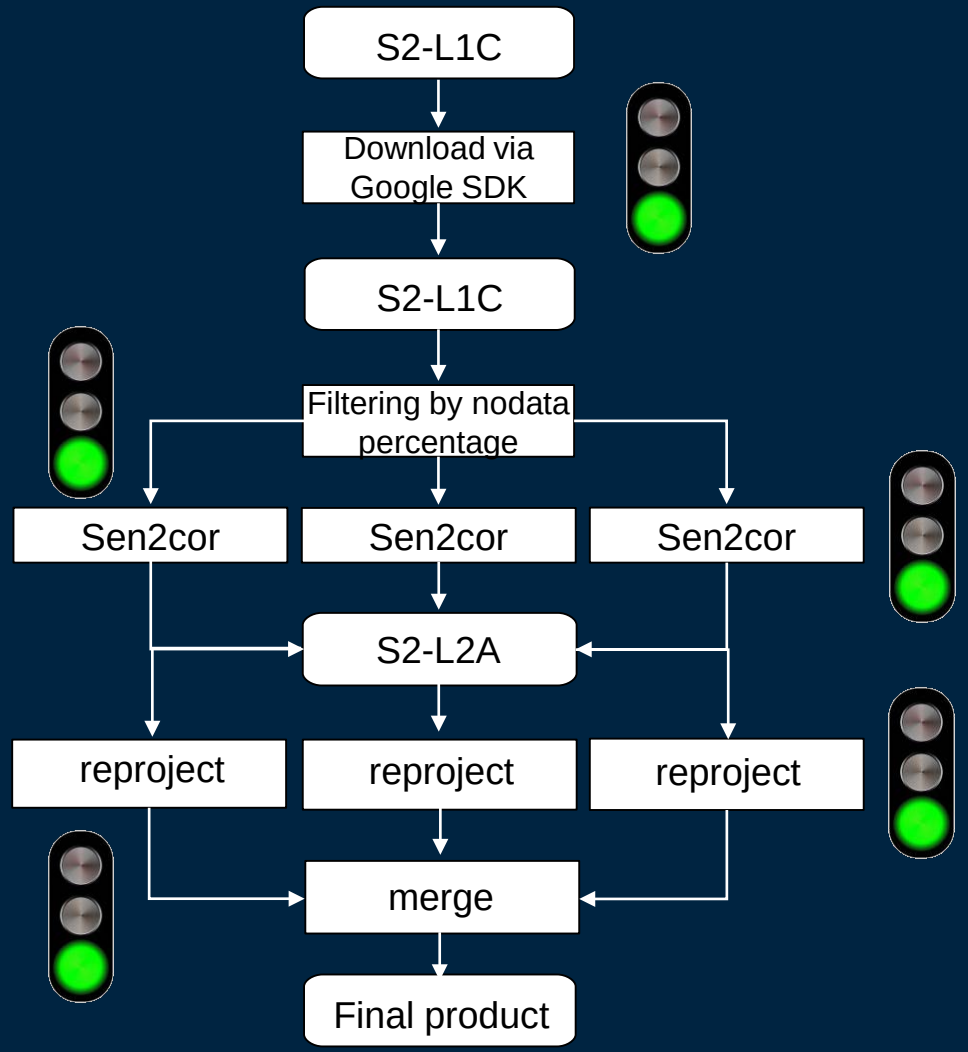


Recommendations

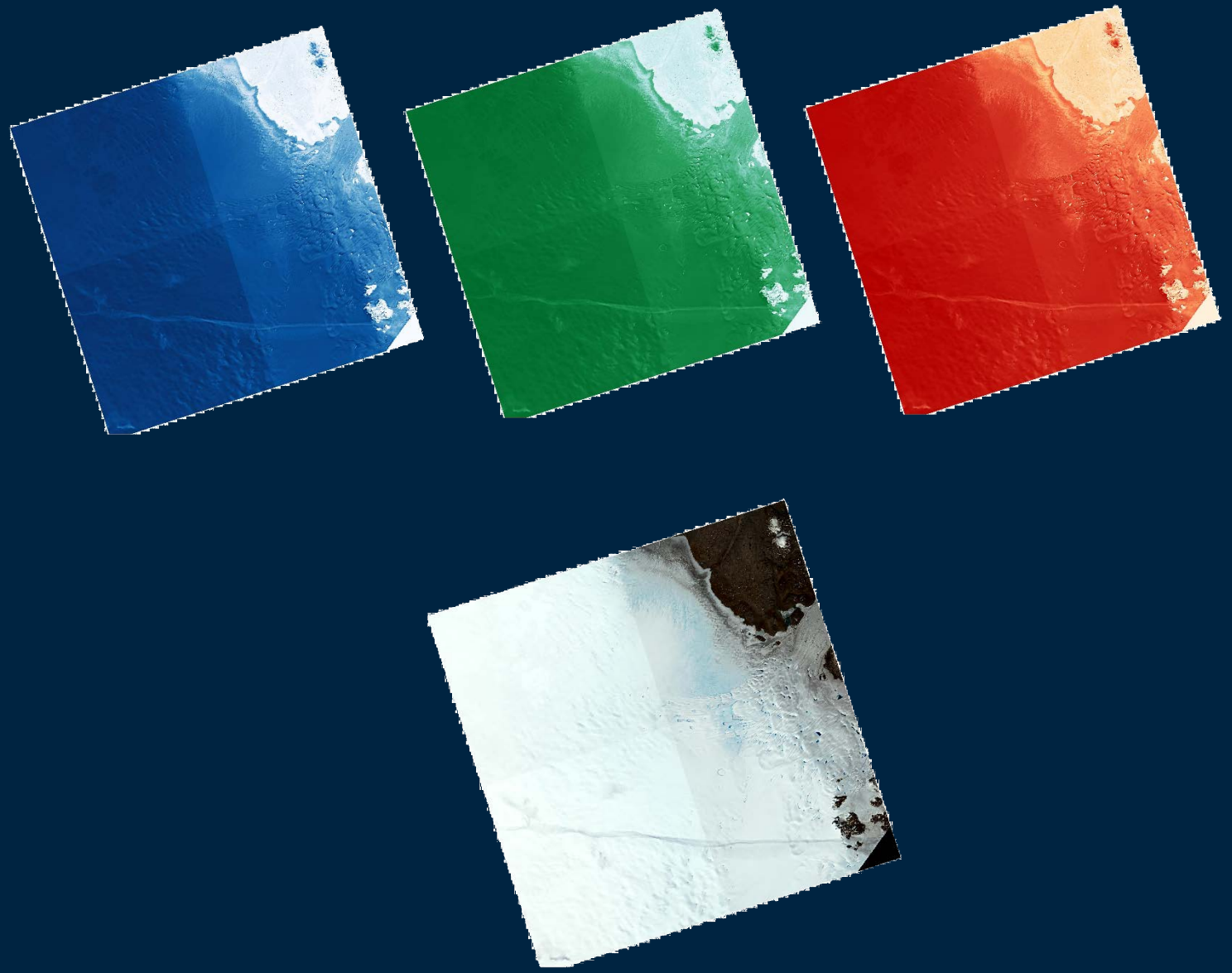
- Threads and RAM: relation 1:4(GB)
- Storage: 600 MB/scene; L2A data factor 1.3 larger than L1C
  → at least double the initial L1C storage necessary

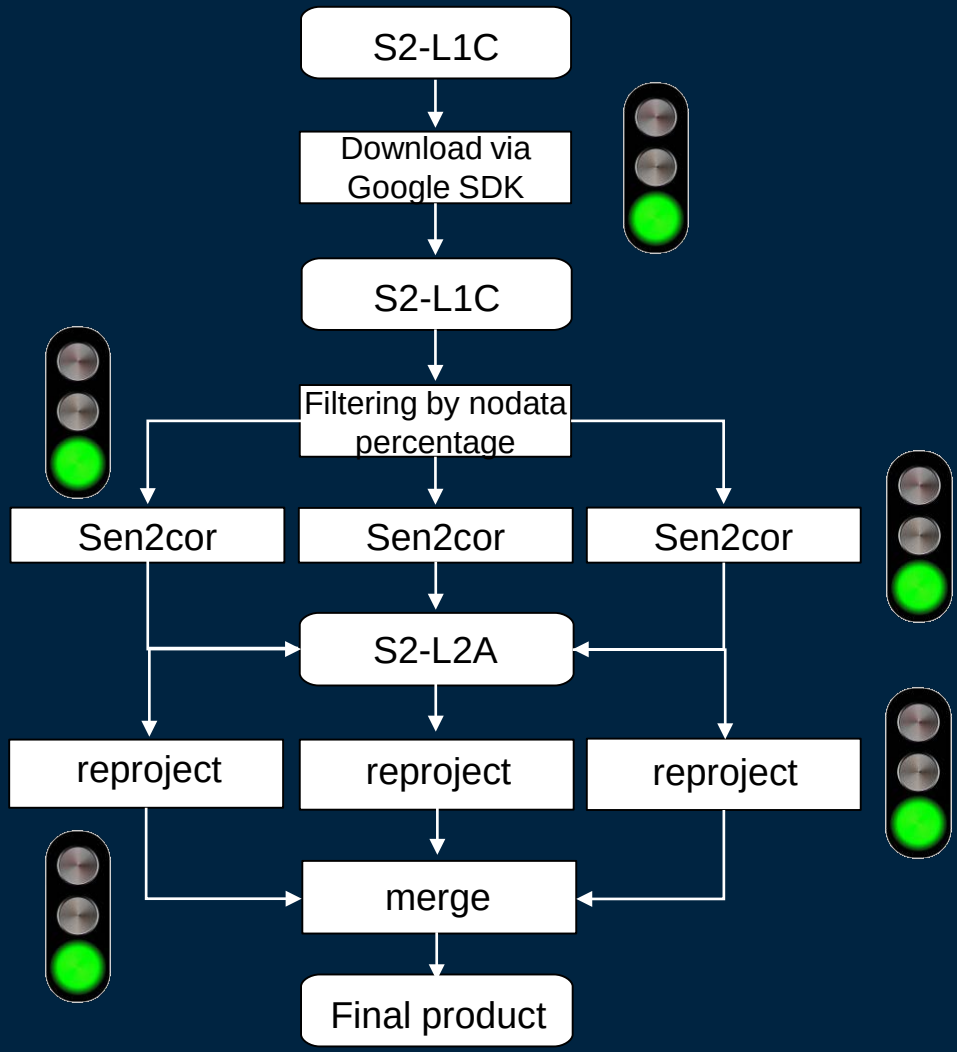
Caveats

- Results still contain cloudy data
  → subsequent selection based on SCL
- Still potential for optimization, e.g. parallelized GC downloads



Take home




S2-L1C
Download via Google SDK
S2-L1C
Filtering by nodata percentage
Sen2cor
Sen2cor
Sen2cor
S2-L2A
reproject
reproject
reproject
merge
Final product

Aims:

- Develop a processing chain that:
  - Uses local ressources
  - Is based on free & open source software alone
  - Is scalable based on initial knowledge of AOI size
  - Is fully automated
- Identify bottlenecks in the processing chain and possible solutions
- Give hardware recommendations based on empirical knowledge

ubuntu

R

GDAL

esa

Sen2Cor Processor

Thanks for your attention!

Sen2Cor Processor