Remote Sensing -Deployable Analysis Environment

Power your workflow with HPC

Meiert W. Grootes, Utrecht 23-01-2024

The RS-DAT team

NLeSC:

Team Atlas (Pranav Chandramouli, Meiert Grootes, Ou Ku, Francesco Nattino, Fakareh (Sarah) Alidoost, Yifat Dzigan)

SURF:

Ander Astudillo, Martin Brandt, Natalie Danezi, Robert Griffioen, Annette Langedijk, Raymond Oonk

netherlands eScience center

SURF

netherlands eScience center

The Netherlands eScience Center

“Empowering researchers across all disciplines through advanced research software”

National centre / independent foundation since 2012 /

NWO & SURF

netherlands eScience center

MISSION

empowering researchers across all disciplines through innovative research software.

AMBITION 1
collaboratively designing software for research

AMBITION 2
building digital expertise

STRATEGIC PRIORITIES

- ✔ institutional alignment
- ✔ software sustainability

- open calls for eScience domain research
- collaborations in advanced eScience technologies
- workshops & training
- community building & networking

netherlands eScience center

Let’s stay in touch

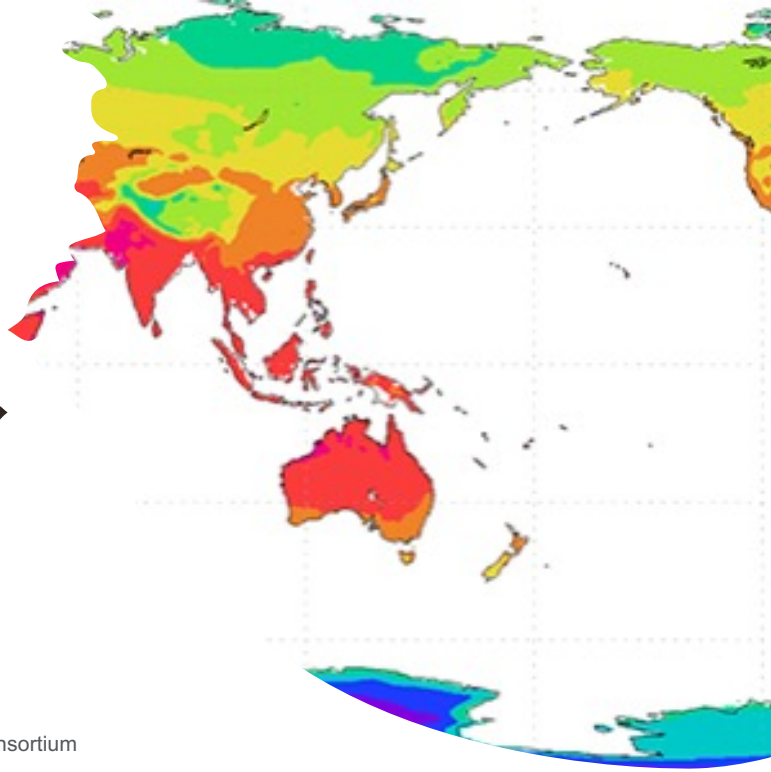
Check for our open calls

www.eScienceCenter.nl

info@esciencecenter.nl

+31 20 460 4770

ESA Euclid consortium

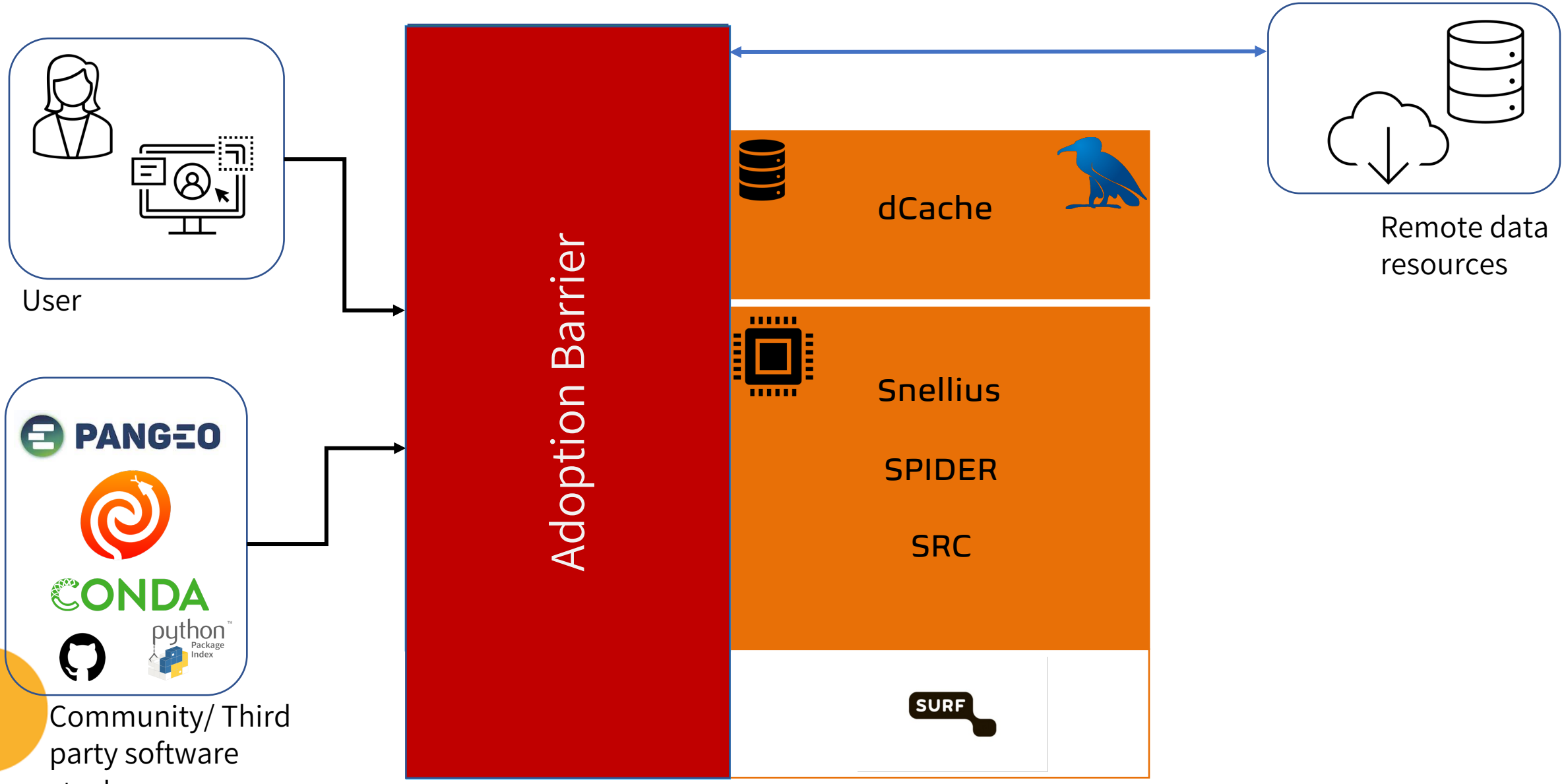


ESA/Hubble & NASA, ESO/ Lutz Wisotzki et al.

Democratizing Big Data

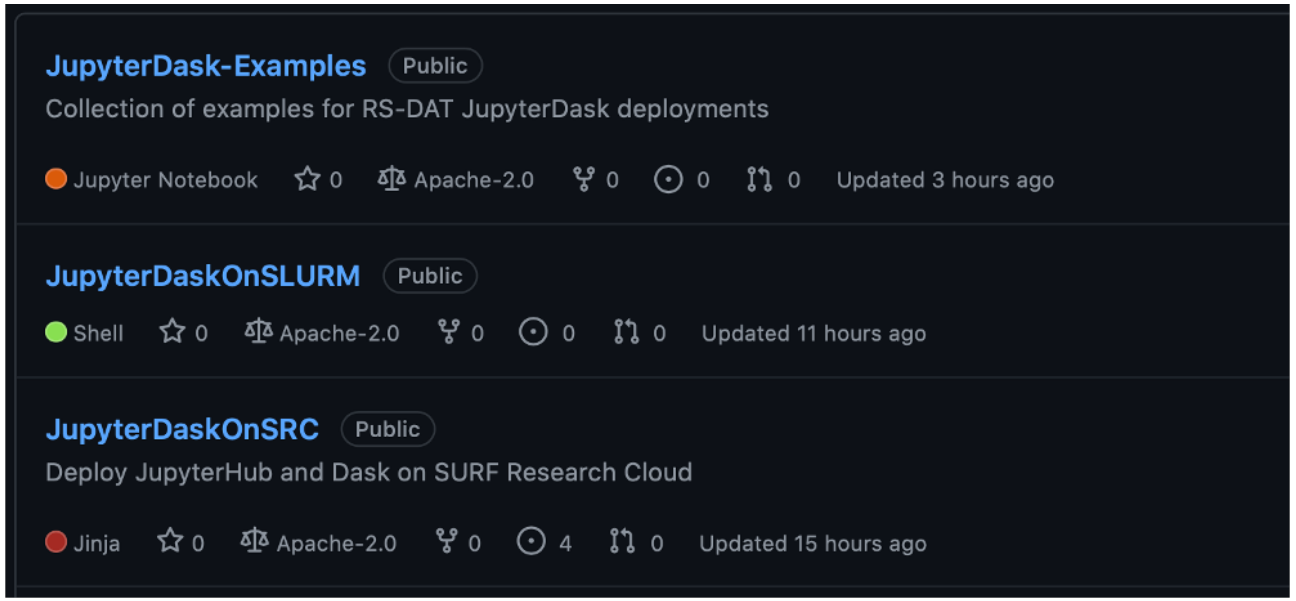
- Big Data ubiquitous across EO and astronomy (EO/astro sats, imaging surveys, large FOV IFUs, sims)
- Fundamental challenge – storage & processing
- Low barrier (local) processing solutions do not scale
- Big data solutions often limited by upfront investment (time/money) and steep learning curve

What is RS-DAT ?



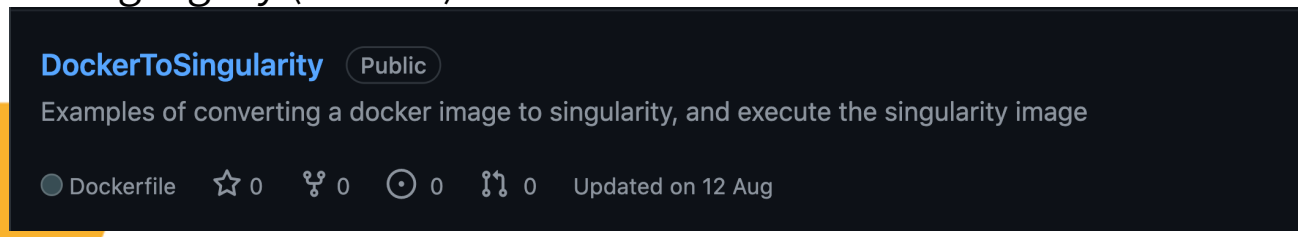
RS-DAT Components

https://github.com/RS-DAT/JupyterDaskOnSLURM



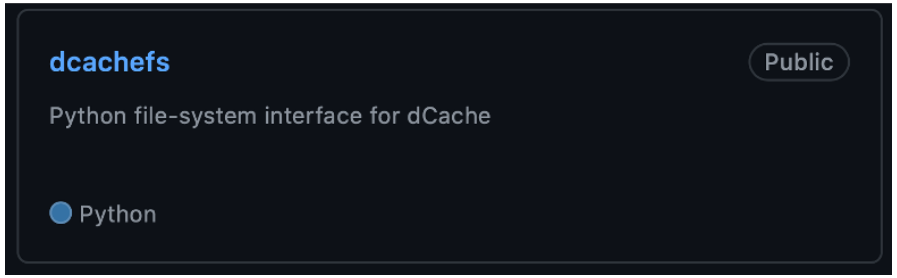
Scalable analysis with Jupyter

Using legacy (Docker) containers for HPC



https://github.com/RS-DAT/DockerToSingularity

https://github.com/NLeSC-GO-common-infrastructure/dcachefs



Python interface to dCache

https://github.com/NLeSC-GO-common-infrastructure/stac2dcache

utility functions to manage STAC catalogs (and the underlying data) on dCache.

Data storage and access - dCache

dCache

- Mass storage system supporting heterogenous nodes as single virtual filesystem
- Multiple I/O, incl. HTTP/Webdav
- Powerful system but not fully integrated with (python) analysis workflows

dCacheFS

- Python-based, extending Filesystem Spec (`fsspec`).
- Mainly, one filesystem class following `fsspec`'s API (implementing functions like `ls`, `rm`, `cat`, etc.).
- File like objects to read with other libraries
- Automatic integration with other libraries using `fsspec`

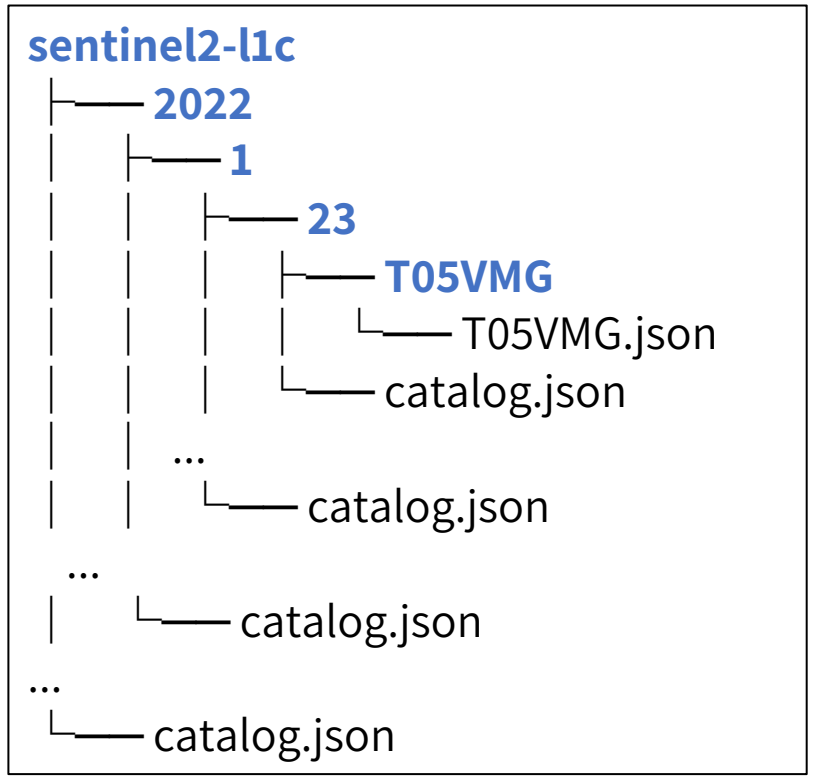
```
import dcachefs
import fsspec

with fsspec.open('https://url/to/myfile.tif', 'r') as fr:
    with fsspec.open('dcache://path/to/myfile.tif', 'w') as fw:
        fw.write(fr.read())
```

STAC and STAC2dCache

Spatio-Temporal Asset Catalogue

- Common structure to describe spatio-temporal data.
- Standard way to catalog geospatial data files.
- All about metadata, linked to data.
- STAC ecosystem includes:
  - Specifications (core elements definitions);
  - API;
  - Tools.



netherlands eScience center

STAC2dCache

- STAC2dCache extends the I/O functionality of core STAC library (PySTAC) to read/write metadata from/to the dCache storage.
- Provides utility functions to copy data files to/from dCache

```python
import stac2dcache
from pystac import Catalog

catalog = Catalog(...)
catalog.add_items(...)
catalog.normalize_and_save(
    'dcache://path/to/catalog'
)
```

```python
from stac2dcache.utils import copy_asset, get_asset

catalog = Catalog.from_file('dcache://path/to/catalog.json')

for asset_key in ('red', 'green', 'blue'):
    copy_asset(catalog, asset_key, update_catalog=True)

da = get_asset(catalog, 'blue', 'T05VMG')
da.plot()
```

netherlands eScience center

A (Big) Data analysis ecosystem

Pangeo is first and foremost a community of people working collaboratively to develop software and infrastructure to enable Big Data geoscience research.

Some of the products produced by this community include interconnected software package and deployments of this software in cloud and high-performance-computing environments. Such a deployment is sometimes referred to as a *Pangeo Environment*.

PANGEO
A community platform for Big Data geoscience

The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

dask

xarray

jupyter

Zarr

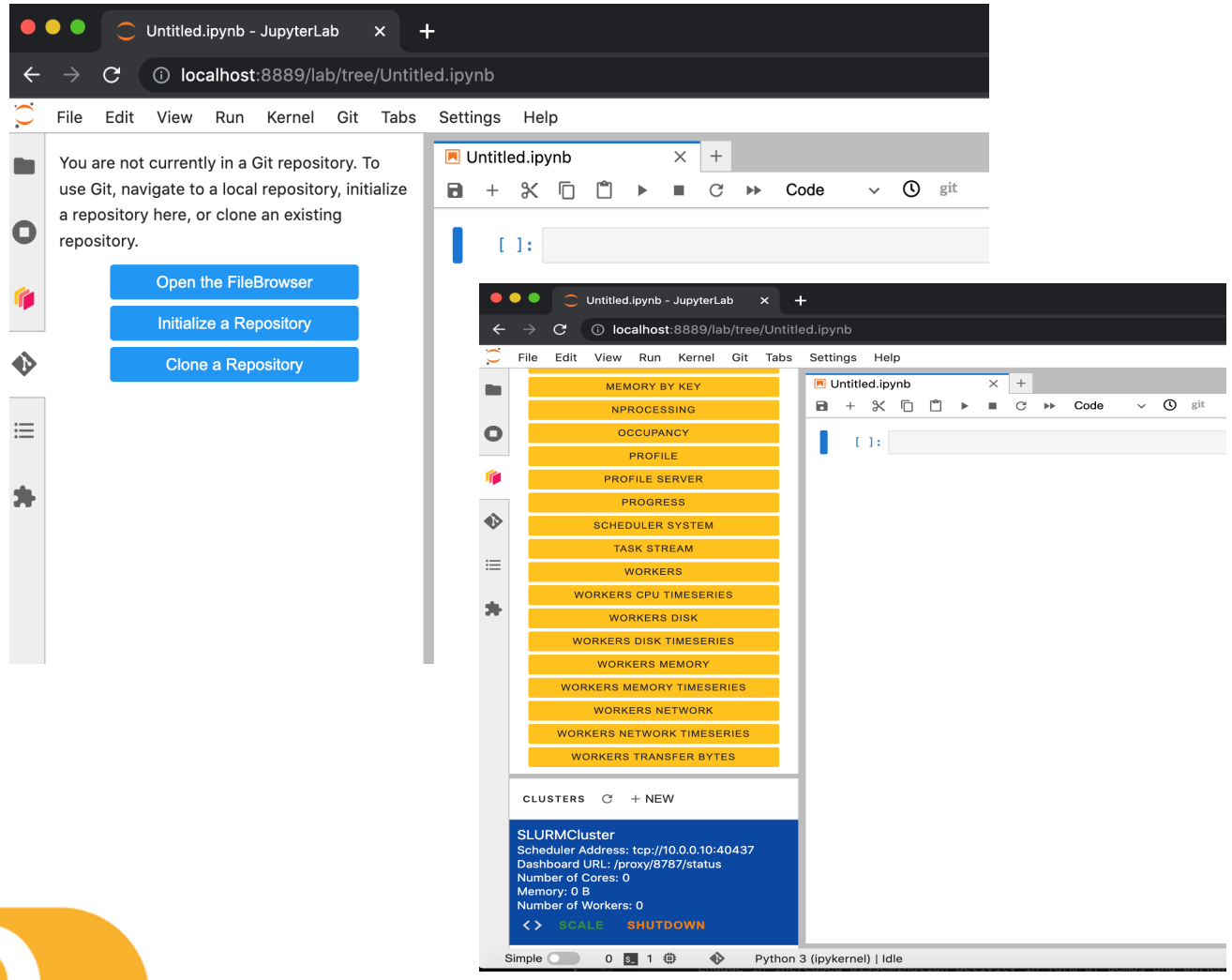
Scalable computation

Rich data model w/ out-of-core support

Interactive analysis and execution

Flexible storage

Scalable analysis - Jupyter at scale

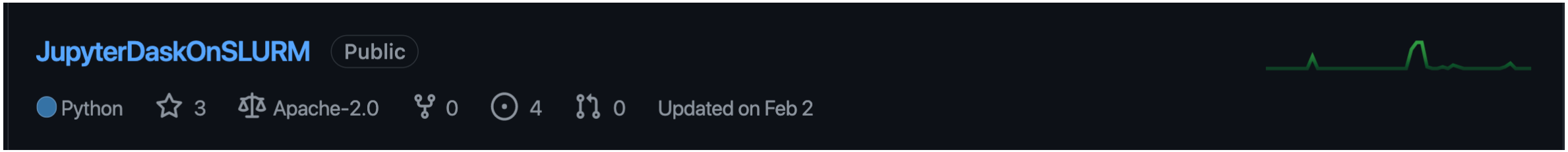


- Scale analysis: accustomed interactive workflows, but backed by HPC/Cloud
- Support for PyData/Python ecosystem
- Combine Jupyter server with Dask cluster on HPC/HTC/Cluster/Cloud system
- Fully integrated with storage (dCache)
- Expose Jupyter python ecosystem (lab plugins …)

netherlands eScience center

Jupyter Dask on Slurm

SURF H*C Infrastructure: Spider (HTC), Snellius (HPC); SLURM scheduler

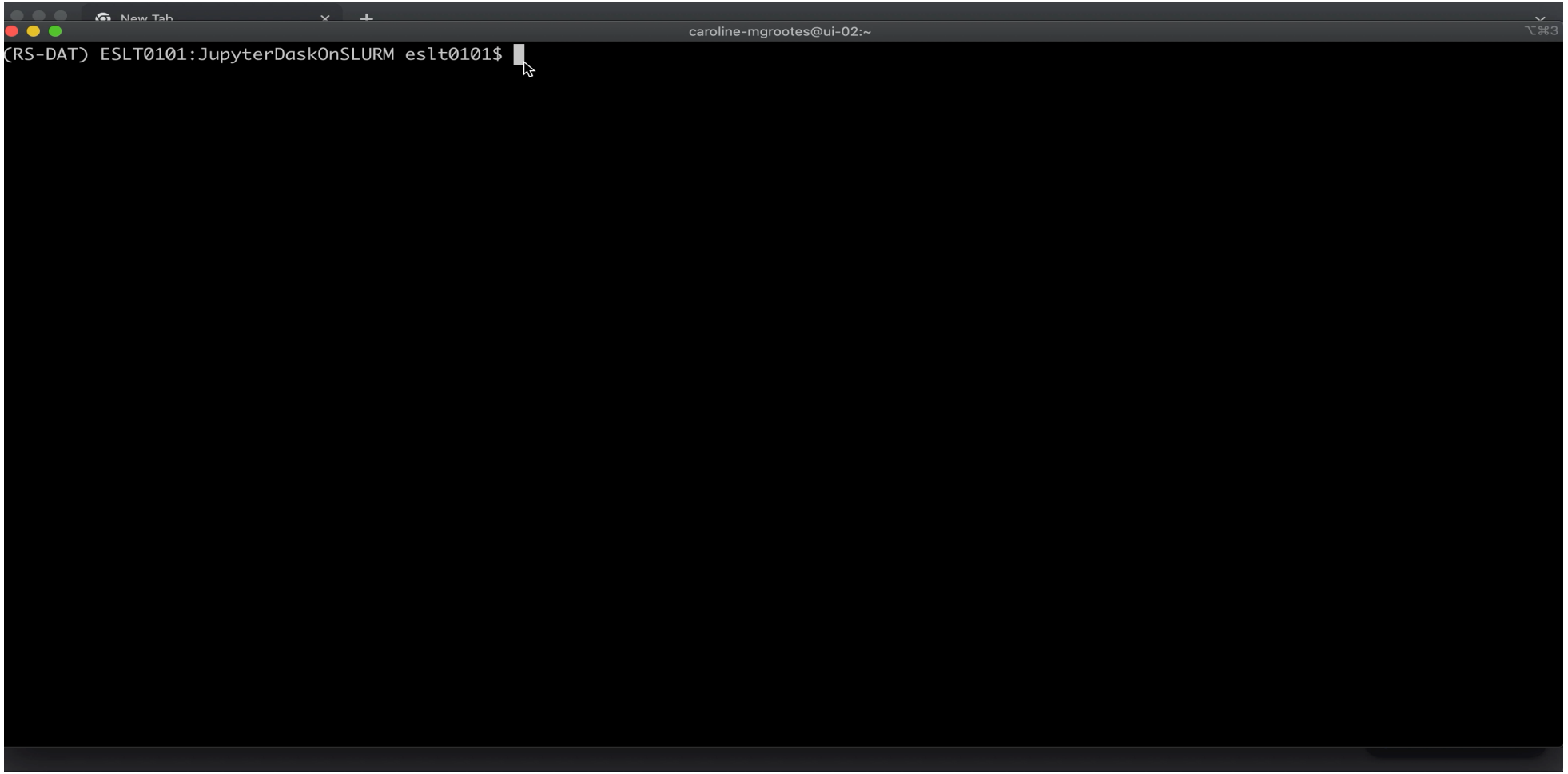


- JupyterLab instance with Dask and Git extensions, scalable Dask cluster, running on SLURM managed HTC/HPC system
- Basic idea
  - Launch JupyterLab server and Dask scheduler as long-running batch job
  - SSH port-forwarding to connect to JupyterLab server
  - Launch workers as short-lived batch jobs (fast thru queue)

Ensure ease of use by abstracting details from user (but configurable if desired)

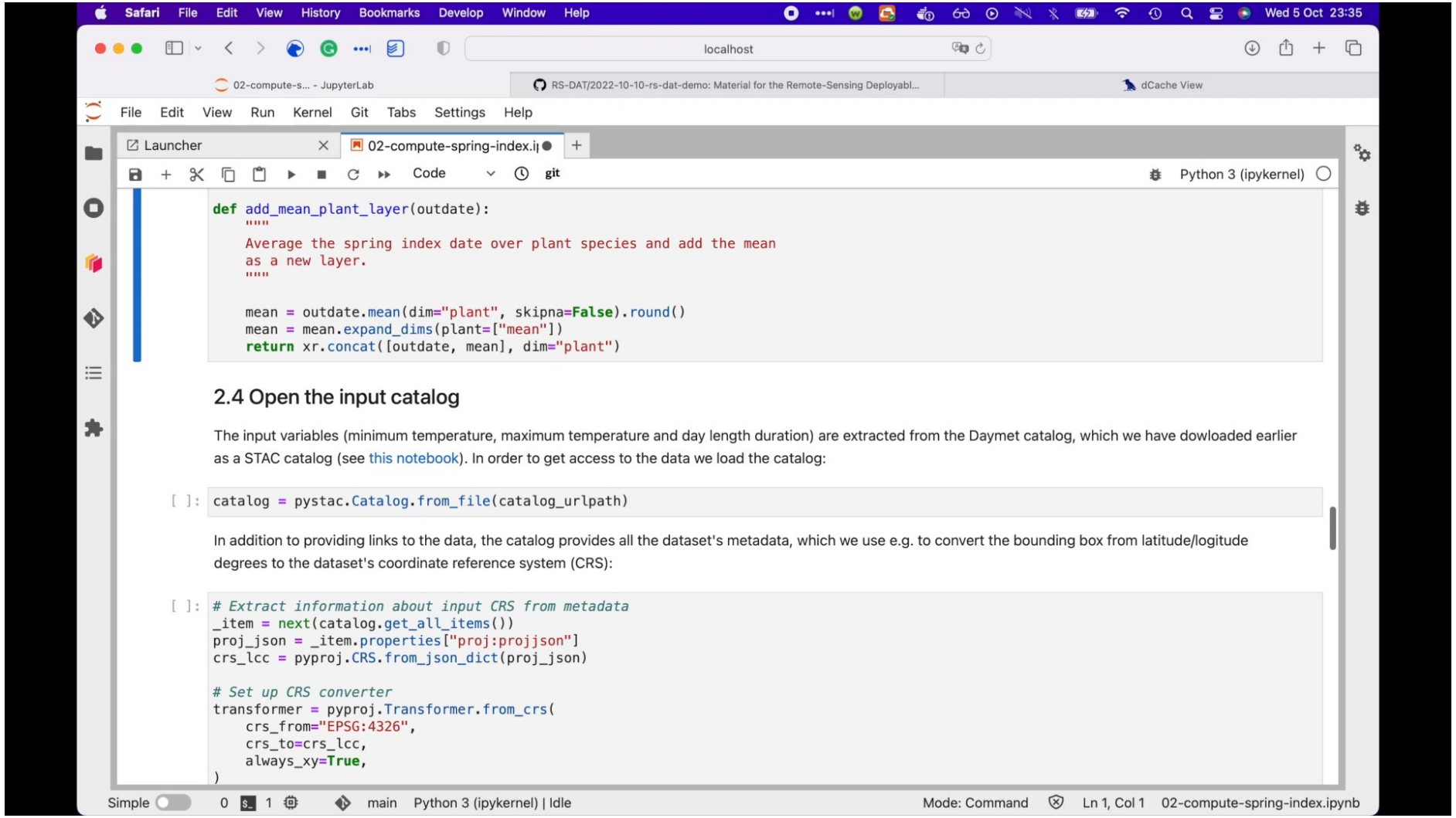
RS-DAT in practice

- Set up your environment.yaml file and you're good to go locally
- Lauch via command line
- That's it!



netherlands eScience center

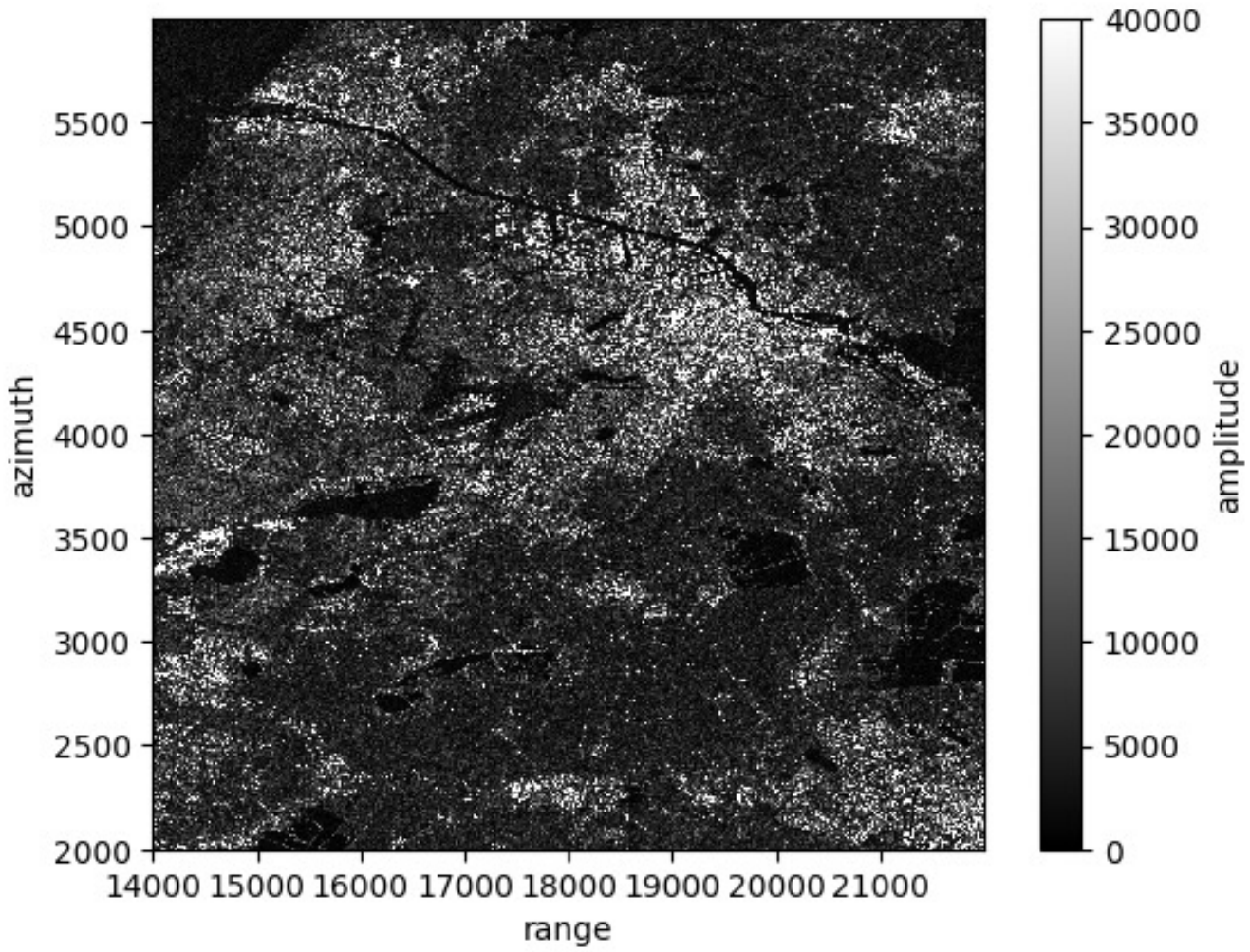
RS-DAT in practice



https://github.com/RS-DAT

netherlands eScience center

RS-DAT off the shelf



Mean Reflectance Map of Amsterdam

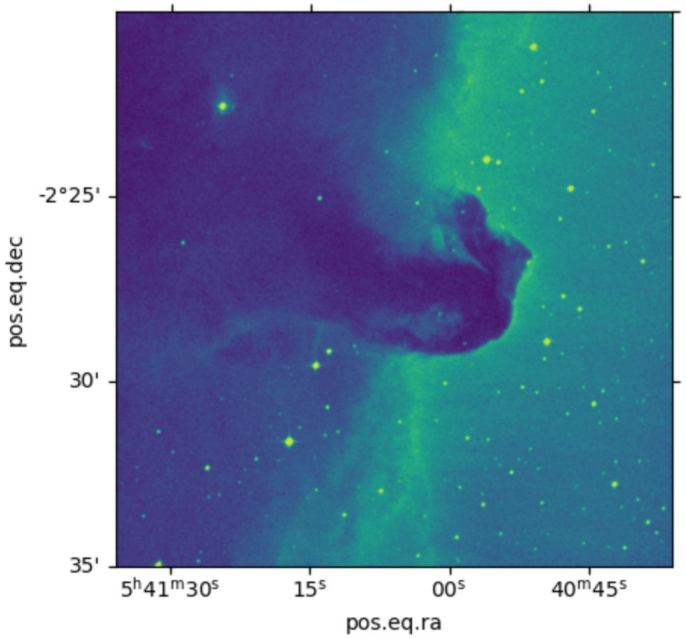
Images processed: 100
Images size: ~500 GB

RS-DAT Runtime: ~10 mins
Previous Runtime: ~1 week

- Off-the-shelf use of DAT
- Integration with SARXarray
- Simple HTC parallelization of established workflow – orders of magnitude speed

netherlands eScience center

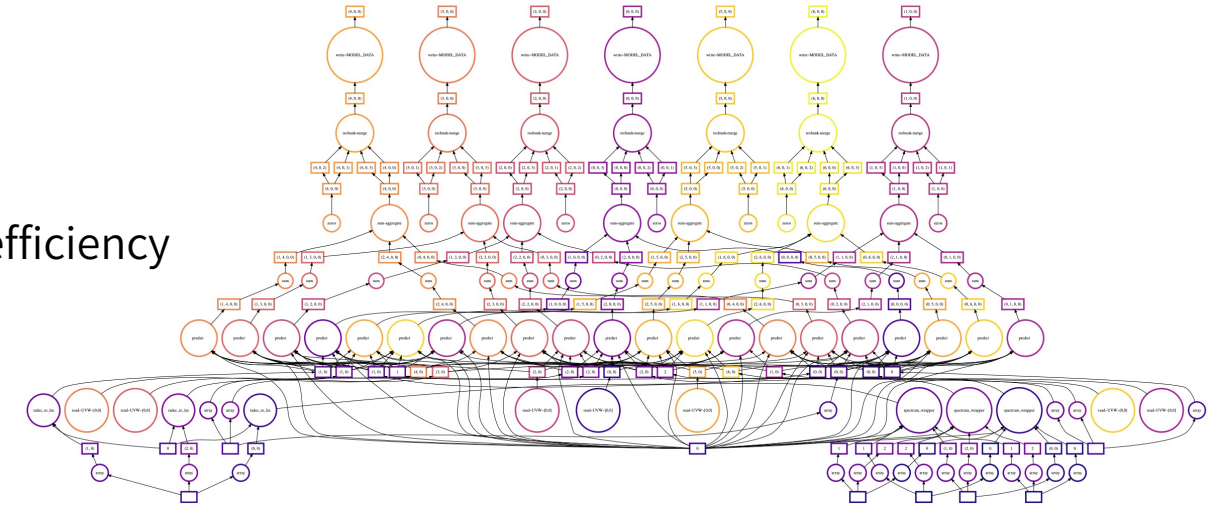
RS-DAT off the shelf



Horse Head nebula from FITS w/ ndcube

- Dask backed image/IFU data processing w/ ndcube
- Large array of (nascent) radio astronomy tooling based on dask; dask-ms, SKA Fourier transform, etc.
- LSDB & HiPsCat; Dask backed astronomical source catalog (cross-)analysis at PB scale for LSST

- Packages are python ecosystem and dask based
- Require significant compute and storage resources for efficiency
- Should work off-the-bat with RS-DAT.
- We be interested and happy to help/think along



Prediction of model visibilities using dask-ms. Perkins et al 2021

netherlands eScience center

Summary

- RS-DAT provides easy to deploy scalable analysis
- Full integration with PyData/Python ecosystem
- Full integration with mass storage dCache
- Fully automated deployment on SURF infrastructure. No further configuration needed. Customized (local) deployments also simple -> DelftBlue
- Work in progress – HPC and SURF research cloud integration, ML for RS support, …
- What do you need? Could you use this?

What would you like/need? Come talk to us!

team-atlas@esciencecenter.nl

netherlands eScience center

Let's stay in touch!

www.eScienceCenter.nl

m.grootes@esciencecenter.com

+31 (0)20 460 4770