Earth System Data Cube Version 2.0.0

This repository contains the scripts, configurations as well as providers used for individual Earth System Data Cube (ESDC) production of version 2.0.0. The repository also contains md5 sums of the released data.

Meta Info

- ESDL Cube 1/4 degree
- ESDL Cube 1/12 degree
- ESDL Cube variables 1/4 as well as 1/12 degree

Usage examples

Also use 2018_12_03_cube_usage_examples.ipynb

Open a Cube (config: 8d, 0.25deg)

```
import xarray as xr
ds = xr.open_zarr('/home/jovyan/work/datacube/ESDCv1.0.2_2/esdc-8d-0.25deg-1x720x1440-1.0.2_
ds
```

Open a variable (e.g. par)

```
ds.par
```

Plot a time step

```
v = ds.par.sel(time='2003-01-05')
v.plot()
```

Plot a regional subset:

```
v = ds.par.loc[dict(lat=slice(40, -20), lon=slice(-20, 20), time='2003-01-05')]
v.plot()
```

Plot a time step close to your choice

```
v = ds.par.sel(time='2003-01-08', method='nearest')
v.plot()
```

Plot a time series of a cell close to a spatial location

```
v = ds.par.sel(dict(lat=51, lon=10), method='nearest')
v.plot()
```

Cube generation

Installing Pre-requisites for the Cube Generation

- Miniconda
  - Install Miniconda
- Install cate
  - `git clone https://github.com/esa-esdl/cube-generator.git`
  - `cd [cube-generator]/cate`
  - `conda env create --file=environment.yml`
  - `source activate cate` (Linux) or `activate esdl` (Windows)
  - `python setup.py  install`
- Install esdl-core
  - `cd [cube-generator]/esdl-core`
  - `python setup.py install`
  - Run `cuge-gen` to test whether the software is correctly installed

Providers

The esdl core library contains data providers already. However, for this version we developed new providers for new and updated variables. They are stored in the sub-directory `cube/providers`. In order to activate them, you have to run `python setup.py` from the cube-generator root directory.

Adding and updating Variables for v1.0.2_2

1. `cd [cube-generator]/scripts`
2. rsync down cube
3. check md5sums
4. `./cube-gen.sh`
5. Copy new variables to cube
6. Generate md5sums of new variable
7. rsync cube up
8. commit new md5sums to repo

The cube version 1.0.2_2 uses the zarr data format. Zarr serialises chunked data and, thus, is comprised of a large number of small files for each variable.

The following script will generate a total md5sums for each directory.

```
bash scripts/checksum.sh -c [cube]
```

This script will check the integrity of the md5sums across a cube:

```
bash check_data_integrity.sh -c [cube]
```

Naming convention

The following naming convention for the cube directories has been used: https://github.com/esa-esdl/cube-generator/wiki/Cube-file-structure-and-naming-convention