



Climate
Change

A COMMON DATA MODEL APPROACH TO NETCDF AND GRIB DATA HARMONISATION

Alessandro Amici, B-Open, Rome

@alexamici - <http://bopen.eu>

Workshop on developing Python frameworks for earth system sciences, 2017-11-28, ECMWF, Reading.

MOTIVATION: FORECAST DATA AND TOOLS

- ECMWF
 - Weather forecasts, re-analyses, satellite and in-situ observations
 - N-dimensional gridded data in GRIB
 - Archive: Meteorological Archival and Retrieval System (MARS)
 - A lot of tools: Metview, Magics, ecCodes...

MOTIVATION: CLIMATE DATA

- Copernicus Climate Change Service (C3S / ECMWF)
 - Re-analyses, seasonal forecasts, climate projections, satellite and in-situ observations
 - N-dimensional gridded data in many dialects of NetCDF and GRIB
 - Archive: Climate Data Store (CDS)



Climate
Change

HARMONISATION STRATEGIC CHOICES

Metview Python Framework and CDS Toolbox projects

- Python 3 programming language
 - scientific ecosystems
- xarray data structures
 - NetCDF data model: variables and coordinates
 - support for arbitrary metadata
 - CF Conventions support on IO
 - label-matching broadcast rules on coordinates
 - parallelized and out-of-core computations with dask

ECMWF NETCDF DIALECT

```
>>> import xarray as xr
>>> ta_era5 = xr.open_dataset('ERA5-t-2016-06.nc', chunks={}).t
>>> ta_era5
<xarray.DataArray 't' (time: 60, level: 3, latitude: 241, longitude: 480)>
dask.array<open_dataset-..., shape=(60, 3, 241, 480), dtype=float64, chunksize=(60, 3,
Coordinates:
  * longitude  (longitude) float32 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 ...
  * latitude  (latitude) float32 90.0 89.25 88.5 87.75 87.0 86.25 85.5 ...
  * level     (level) int32 250 500 850
  * time      (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 ...
Attributes:
  units:          K
  long_name:      Temperature
  standard_name:  air_temperature
>>> ta_era5.level.attrs['units']
'millibars'
```

CMIP5 NETCDF DIALECT

```
>>> ta_cmip5 = xr.open_dataset('ta_6hrPlev_CMCC-CM_decadal2005_2017-06.nc', chunks={}).
>>> ta_cmip5
<xarray.DataArray 'ta' (time: 120, plev: 3, lat: 240, lon: 480)>
dask.array<open_dataset-..., shape=(120, 3, 240, 480), dtype=float64, chunksize=(120, 3
Coordinates:
  * time      (time) datetime64[ns] 2017-06-01 2017-06-01T06:00:00 ...
  * plev      (plev) float64 8.5e+04 5e+04 2.5e+04
  * lat       (lat) float64 -89.43 -88.68 -87.94 -87.19 -86.44 -85.69 -84.95 ...
  * lon       (lon) float64 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 6.75 7.5 ...
Attributes:
  standard_name:      air_temperature
  long_name:          Air Temperature
  units:              K
  original_name:      t
  cell_measures:      area: areacella
  associated_files:   baseURL: http://cmip-pcmdi.llnl.gov/CMIP5/dataLocation...
  history:            2012-04-13T21:01:28Z altered by CMOR: Inverted axis: lat.
>>> ta_cmip5.plev.attrs['units']
'Pa'
```

INTEROPERABILITY IS HARD

```
>>> ta_era5 - ta_cmip5
<xarray.DataArray (time: 60, level: 3, latitude: 241, longitude: 480, plev: 3, lat: 240)
dask.array<sub, shape=(60, 3, 241, 480, 3, 240, 480), dtype=float64, chunksize=(60, 3,
Coordinates:
  * time          (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 ...
  * longitude     (longitude) float32 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 ...
  * latitude      (latitude) float32 90.0 89.25 88.5 87.75 87.0 86.25 85.5 ...
  * level         (level) int32 250 500 850
  * plev          (plev) float64 8.5e+04 5e+04 2.5e+04
  * lat           (lat) float64 -89.43 -88.68 -87.94 -87.19 -86.44 -85.69 ...
  * lon           (lon) float64 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 6.75 ...
```






INTEROPERABILITY IS HARD

```
>>> ta_era5.rename({'latitude': 'lat', 'longitude': 'lon', 'level': 'plev'}) - ta_cmip5
<xarray.DataArray (time: 60, plev: 0, lat: 0, lon: 480)>
dask.array<sub, shape=(60, 0, 0, 480), dtype=float64, chunksize=(60, 0, 0, 480)>
Coordinates:
  * time      (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 2017-06-02 ...
  * plev      (plev) object
  * lat       (lat) float64
  * lon       (lon) float32 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 6.75 7.5 ...
```




HARMONIZATION BEYOND XARRAY

- xarray . DataArray data structure
 - GRIB as first class citizen
 - coordinates harmonisation
 - matching labels, units and centre values (e.g same grid)
 - more advanced use of metadata / CF Conventions

ECMWF HARMONISATION PROJECTS

- pythonisation of ECMWF data and tools
 - xarray GRIB driver (uses ecCodes) 
 - high level python bindings
 - Metview , Magics 
 - low level python bindings
 - ecCodes , odb-api 

C3S / ECMWF HARMONISATION PROJECTS

- CDS common data model 
 - definition, importer and compliance checker
- CF Conventions aware variables operations 
 - use metadata for units conversion, plot labeling...
 - add metadata to results, update attributes (e.g. units), track provenance...

XARRAY-GRIB-DRIVER: GRIB AS FIRST CLASS CITIZEN

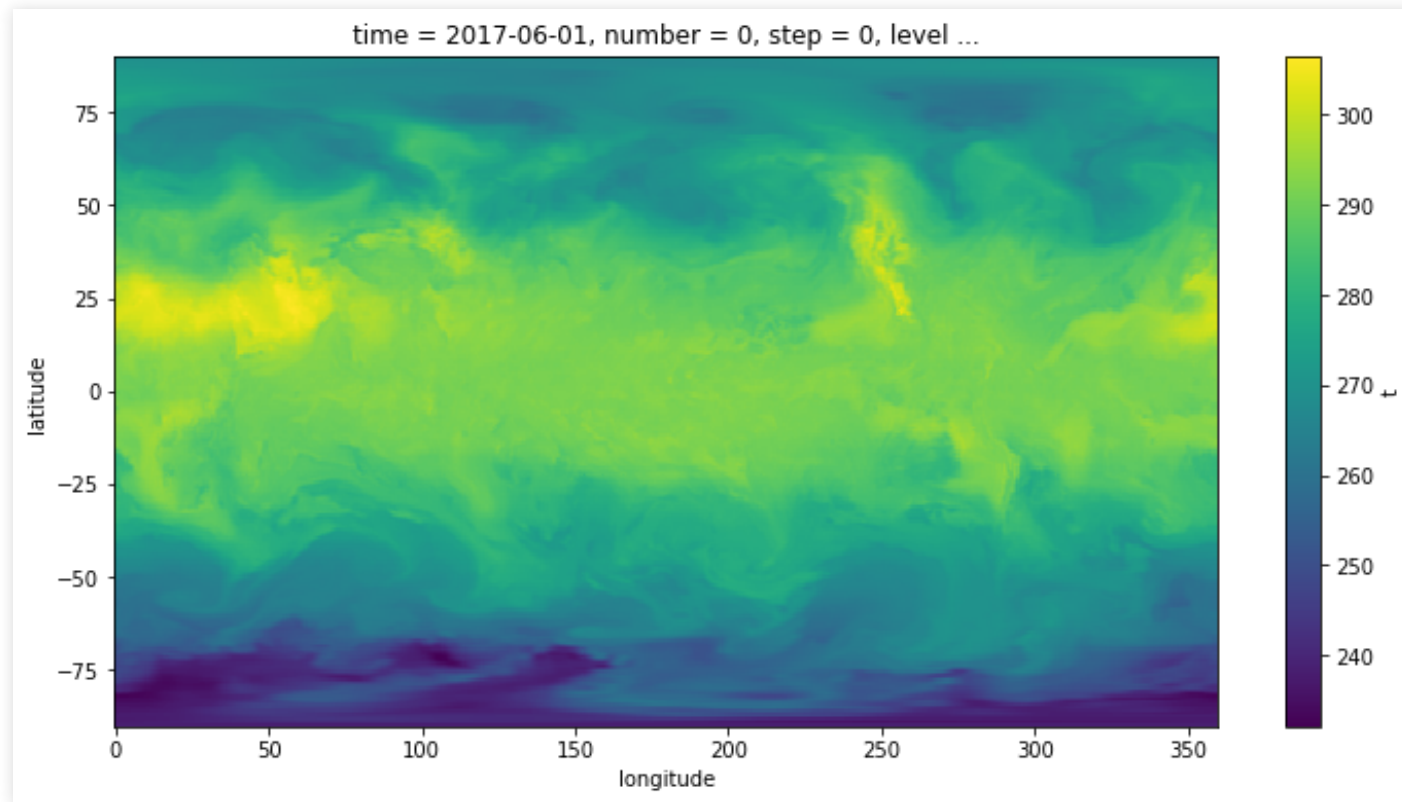
- xarray GRIB driver
 - uses ecCodes
 - low-level GRIB driver similar to netcdf4-python
 - high-level actual xarray driver
 - not public yet, to be release as Open Source
 - pull request to xarray! (at some point)

XARRAY-GRIB-DRIVER: GRIB IN XARRAY

```
>>> import xarray_grib
>>> store = xarray_grib.GribDataStore('ERA5-t-2016-06.grib')
>>> ta_era5_grib = xr.open_dataarray(store)
>>> ta_era5_grib
<xarray.DataArray 't' (time: 60, number: 1, step: 1, level: 3, latitude: 241, longitude: 20822400 values with dtype=float32)
Coordinates:
  * time          (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 ...
  * number       (number) int32 0
  * step         (step) int32 0
  * level        (level) float64 850.0 500.0 250.0
  * latitude     (latitude) float64 90.0 89.25 88.5 87.75 87.0 86.25 85.5 ...
  * longitude    (longitude) float64 0.0 0.75 1.5 2.25 3.0 3.75 4.5 5.25 6.0 ...
Attributes:
  long_name:    Temperature
  units:       K
>>> ta_era5_grib.level.attrs['units']
'hPa'
```

XARRAY-GRIB-DRIVER: GRIB IN XARRAY

```
>>> ta_era5_grib.sel(time='2017-06-01T00:00:00', level=850).plot()
```



XARRAY-GRIB-DATA: ROADMAP

- now:
 - internal alpha release
- beginning 2018:
 - public beta release
- mid 2018:
 - propose integration in xarray main repo

CDS-CMOR-TABLES: CDS CDM DEFINITION

- CDS Common Data Model:
 - based on CMIP6 and CDS seasonal forecasts
 - Climate Model Output Rewriter (CMOR) definition files
 - compliance checker tool
 - simple configurable import tool
 - not public yet, decision about open-sourcing the project is underway

CDS-CMOR-TABLES: IMPORT ECMWF

```
$ make_cdscdm ERA5-t-2016-06.nc -o ERA5-t-2016-06_ta_cdm.nc
```

```
>>> ta_era5_cdm = xr.open_dataarray('ERA5-t-2016-06_ta_cdm.nc')
>>> ta_era5_cdm
<xarray.DataArray 'ta' (time: 60, plev: 3, lat: 241, lon: 480)>
[20822400 values with dtype=float32]
Coordinates:
  * lat      (lat) float64 -90.0 -89.25 -88.5 -87.75 -87.0 -86.25 -85.5 ...
  * plev     (plev) float64 8.5e+04 5e+04 2.5e+04
  * time     (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 2017-06-02 ...
  * lon      (lon) float64 -180.0 -179.2 -178.5 -177.8 -177.0 -176.2 -175.5 ...
Attributes:
  long_name:      temperature
  standard_name:  air_temperature
  units:          K
```

CDS-CMOR-TABLES: IMPORT CMIP5

```
$ make_cdscdm ta_6hrPlev_CMCC-CM_decadal2005_2017-06.nc \  
-o ta_6hrPlev_CMCC-CM_decadal2005_2017-06_ta_cdm.nc
```

```
>>> ta_c mip5_cdm = xr.open_dataarray('ta_6hrPlev_CMCC-CM_decadal2005_2017-06_ta_cdm.nc')  
>>> ta_c mip5_cdm  
<xarray.DataArray 'ta' (time: 120, plev: 3, lat: 240, lon: 480)>  
[41472000 values with dtype=float32]  
Coordinates:  
* time      (time) datetime64[ns] 2017-06-01 2017-06-01T06:00:00 ...  
* plev      (plev) float64 8.5e+04 5e+04 2.5e+04  
* lat       (lat) float64 -89.43 -88.68 -87.94 -87.19 -86.44 -85.69 -84.95 ...  
* lon       (lon) float64 -180.0 -179.2 -178.5 -177.8 -177.0 -176.2 -175.5 ...  
Attributes:  
  long_name:      temperature  
  standard_name:  air_temperature  
  units:          K
```

CDS-CMOR-TABLES: SIMPLE ANOMALY

```
>>> ta_era5_cdm - ta_cmip5_cdm
<xarray.DataArray 'ta' (time: 60, plev: 3, lat: 0, lon: 480)>
array([], shape=(60, 3, 0, 480), dtype=float32)
Coordinates:
  * time      (time) datetime64[ns] 2017-06-01 2017-06-01T12:00:00 2017-06-02 ...
  * lat       (lat) float64
  * plev      (plev) float64 8.5e+04 5e+04 2.5e+04
  * lon       (lon) float64 -180.0 -179.2 -178.5 -177.8 -177.0 -176.2 -175.5 ...
```

Close, but no cigar! lat grids differ :(

CDSTOOLS: CLIMATE TOOLS

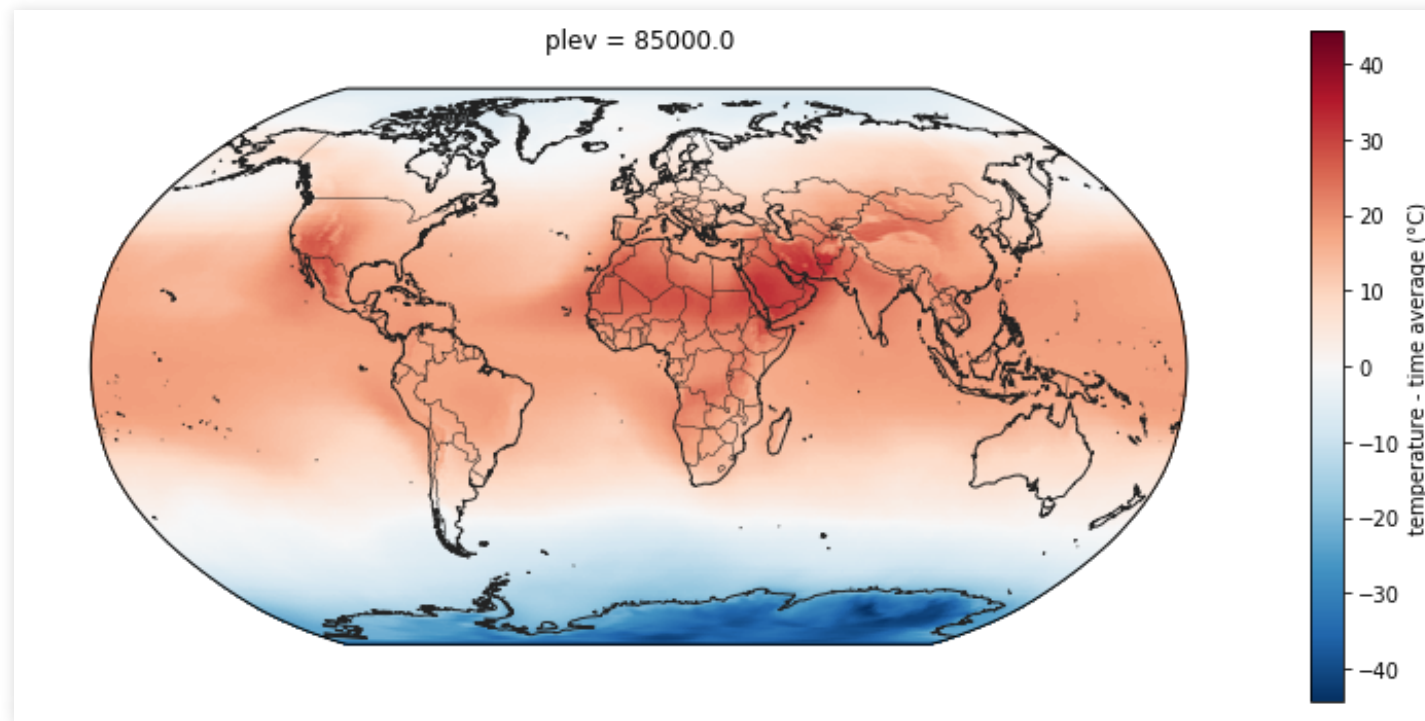
- CDS Tools
 - units conversion during operations
 - internal provenance tracking
 - update of some attributes: e.g. `long_name`, `units`...
 - metadata aware plots: title, legend, labels, units...
 - not public yet, decision about open-sourcing the project is underway

CDSTOOLS: METADATA AWARE OPERATIONS

```
>>> from cdstools import util
>>> ta_era5_low = util.select(ta_era5_cdm, plev=85000)
>>> ta_era5_map = util.average(ta_era5_low, dim='time')
>>> ta_era5_map
<xarray.DataArray 'ta' (lat: 241, lon: 480)>
array([[ 236.114685,  236.114685,  236.114685, ...,  ]], dtype=float32)
Coordinates:
  * lat      (lat) float64 -90.0 -89.25 -88.5 -87.75 -87.0 -86.25 -85.5 ...
    plev     float64 8.5e+04
  * lon      (lon) float64 -180.0 -179.2 -178.5 -177.8 -177.0 -176.2 -175.5 ...
Attributes:
  long_name:      temperature - time average
  standard_name:  air_temperature
  units:          K
```

CDSTOOLS: METADATA AWARE PLOTTING

```
>>> from cdstools import matplotlib as plt
>>> projection = plt.cartopy.crs.Robinson()
>>> fig, ax = plt.subplots(subplot_kw={'projection': projection}, figsize=(12, 6))
>>> plt.geomap(ta_era5_map, ax=ax)
```



TEAMS AND CREDITS

- Coordinators: B. Raoult (ECMWF), S. Siemen (ECMWF), C. Bergeron (C3S / ECMWF) and A. Amici
- Developers and scientific advisers:
 - Metview: I. Russell (ECMWF) and M. Di Bari
 - xarray-grib-driver: A. Amici and L. Barcaroli
 - cds-cmor-tables: C. Cagnazzo, F. Fierli (ISAC-CNR) and G. Cammarota
 - cdstools: G. Cammarota and F. Nazzaro



Climate
Change

THANK YOU!

DO WE HAVE TIME FOR Q/A?

Alessandro Amici, B-Open, Rome

@alexamici - <http://bopen.eu>

Slides: <https://gitpitch.com/alexamici/talks>