missingcoords

February 27, 2025

0.0.1 Missing coordinate attributes after various reductions

When applying any computation involving `apply_ufunc` behind the scenes, the coordinate attributes go missing. The output below shows this for various functions. The expected behaviour is that coordinate `y` always has attributes in the result, while the data itself, `F`, only has attributes if `keep_attrs=True` (except for the dot and cross product).

[1]:
```
import xarray as xr

ds = xr.Dataset(
    data_vars={
        'F':(('x','y'),[[1,2,3],[4,5,6]], {'name':'F'}),
        'w':('x',[1,1],{'name':'weights'}),
    },
    coords={
        'x':('x',[0,1],{'name':'x'}),
        'y':('y',[0,1,2],{'name':'y'}),
    }
)

func = xr.core.duck_array_ops.mean
objs = lambda keep_attrs: [
    ds.F.weighted(ds.w).mean('x', keep_attrs=keep_attrs),
    ds.weighted(ds.w).mean('x', keep_attrs=keep_attrs).F,
    xr.dot(ds.F, ds.w),
    xr.cross(ds.F, ds.w, dim='x'),
    xr.where(ds.x==0, ds.F, ds.w, keep_attrs=keep_attrs),
    xr.apply_ufunc(func, ds.F, input_core_dims=('x',), vectorize=True,␣
  ↪keep_attrs=keep_attrs),
]

print("y attrs:",[obj.y.attrs for obj in objs(keep_attrs=True)])
print("y attrs:",[obj.y.attrs for obj in objs(keep_attrs=False)])
print("attrs:",[obj.attrs for obj in objs(keep_attrs=True)])
print("attrs:",[obj.attrs for obj in objs(keep_attrs=False)])
```

```
y attrs: [{}, {}, {}, {}, {}, {}]
y attrs: [{}, {}, {}, {}, {}, {}]
```

```
attrs: [{'name': 'F'}, {'name': 'F'}, {}, {}, {'name': 'F'}, {'name': 'F'}]
attrs: [{}, {}, {}, {}, {}, {}]
```

None of the above computations above preserve the attributes for `y`. It turns out that `apply_dataarray_vfunc` and `apply_dataset_vfunc` pass on their `keep_attrs` values to `build_output_coord_and_indexes` while this parameter should only affect the data itself, not the coordinates. Setting `combine_attrs` in lines 309,525 of `xr.core.computation.py` to "no_conflicts" fixes all above issues while retaining the proper behaviour for the data (`F`) attributes.

[1]:
```
import xarray as xr

ds = xr.Dataset(
    data_vars={
        'F':(('x','y'),[[1,2,3],[4,5,6]], {'name':'F'}),
        'w':('x',[1,1],{'name':'weights'}),
    },
    coords={
        'x':('x',[0,1],{'name':'x'}),
        'y':('y',[0,1,2],{'name':'y'}),
    }
)

func = xr.core.duck_array_ops.mean
objs = lambda keep_attrs: [
    ds.F.weighted(ds.w).mean('x', keep_attrs=keep_attrs),
    ds.weighted(ds.w).mean('x', keep_attrs=keep_attrs).F,
    xr.dot(ds.F, ds.w),
    xr.cross(ds.F, ds.w, dim='x'),
    xr.where(ds.x==0, ds.F, ds.w, keep_attrs=keep_attrs),
    xr.apply_ufunc(func, ds.F, input_core_dims=('x',), vectorize=True,␣
  ↪keep_attrs=keep_attrs),
]

print("y attrs:",[obj.y.attrs for obj in objs(keep_attrs=True)])
print("y attrs:",[obj.y.attrs for obj in objs(keep_attrs=False)])
print("attrs:",[obj.attrs for obj in objs(keep_attrs=True)])
print("attrs:",[obj.attrs for obj in objs(keep_attrs=False)])
```

```
y attrs: [{'name': 'y'}, {'name': 'y'}, {'name': 'y'}, {'name': 'y'}, {'name':
'y'}, {'name': 'y'}]
y attrs: [{'name': 'y'}, {'name': 'y'}, {'name': 'y'}, {'name': 'y'}, {'name':
'y'}, {'name': 'y'}]
attrs: [{'name': 'F'}, {'name': 'F'}, {}, {}, {'name': 'F'}, {'name': 'F'}]
attrs: [{}, {}, {}, {}, {}, {}]
```

[2]:
```
xr.show_versions()
```

```
/Users/jasperdejong/opt/anaconda3/envs/xarray-tests/lib/python3.10/site-
```

```
packages/pyproj/network.py:59: UserWarning: pyproj unable to set PROJ database
path.
  _set_context_ca_bundle_path(ca_bundle_path)
/Users/jasperdejong/opt/anaconda3/envs/xarray-tests/lib/python3.10/site-
packages/_distutils_hack/__init__.py:30: UserWarning: Setuptools is replacing
distutils. Support for replacing an already imported distutils is deprecated. In
the future, this condition will fail. Register concerns at
https://github.com/pypa/setuptools/issues/new?template=distutils-deprecation.yml
  warnings.warn(


INSTALLED VERSIONS
------------------
commit: None
python: 3.10.16 | packaged by conda-forge | (main, Dec  5 2024, 14:12:04) [Clang
18.1.8 ]
python-bits: 64
OS: Darwin
OS-release: 24.3.0
machine: x86_64
processor: i386
byteorder: little
LC_ALL: None
LANG: nl_NL.UTF-8
LOCALE: ('nl_NL', 'UTF-8')
libhdf5: 1.14.3
libnetcdf: 4.9.2

xarray: 0.1.dev5845+g2475d49
pandas: 2.2.3
numpy: 2.1.3
scipy: 1.15.2
netCDF4: 1.7.2
pydap: 3.5.3
h5netcdf: 1.5.0
h5py: 3.13.0
zarr: 2.18.3
cftime: 1.6.4
nc_time_axis: 1.4.1
iris: 3.11.0
bottleneck: 1.4.2
dask: 2025.2.0
distributed: 2025.2.0
matplotlib: 3.10.0
cartopy: 0.24.0
seaborn: 0.13.2
numbagg: 0.9.0
fsspec: 2025.2.0
```

```
cupy: None
pint: None
sparse: 0.15.5
flox: 0.10.0
numpy_groupies: 0.11.2
setuptools: 75.8.0
pip: 25.0.1
conda: None
pytest: 8.3.4
mypy: 1.15.0
IPython: 8.32.0
sphinx: None
```

[ ]: