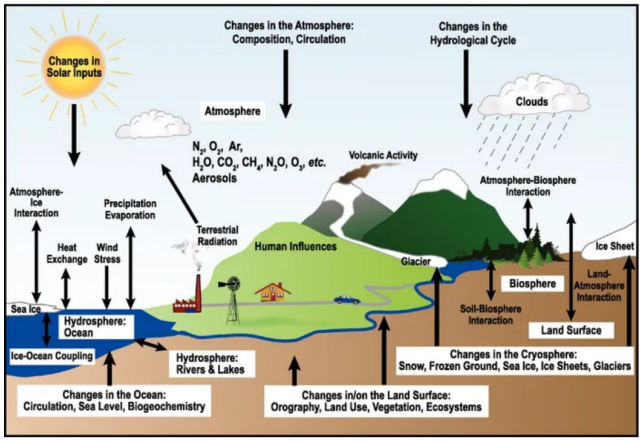
# Scaling UX array:
## Bridging the Gap for High-Performance Unstructured Grid Analysis through Dask and Documentation Enhancements

July 31, 2024

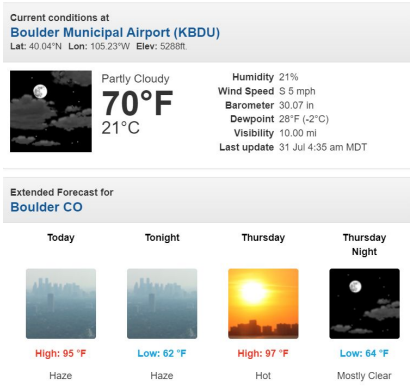
**Rachel Tam**[1,2], Philip Chmielowiec[1], Orhan Eroglu[1]
*Summer Internships in Parallel Computational Science (SIParCS)*
[1] NSF NCAR CISL
[2] University of Illinois, Urbana-Champaign

# Motivation: What are climate models?



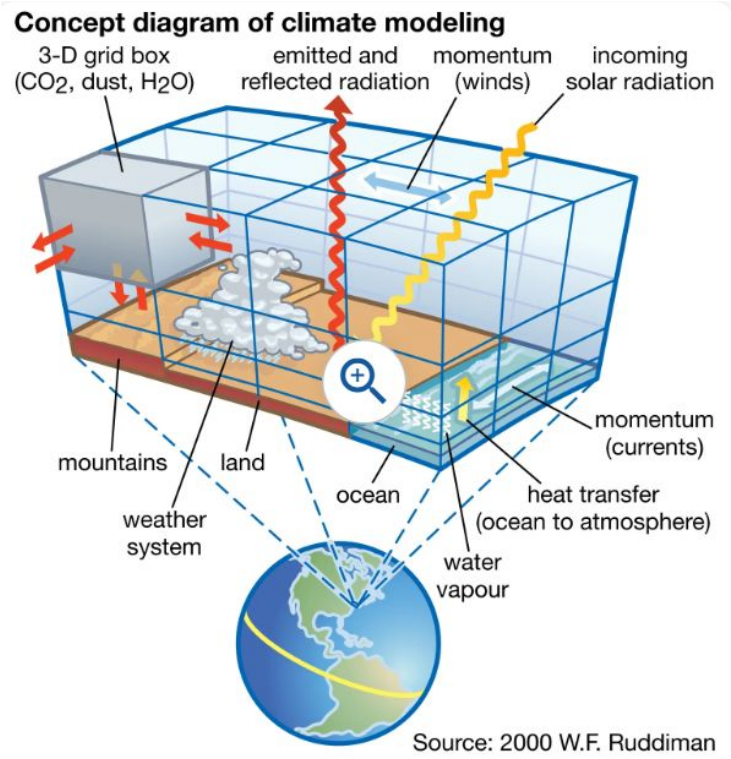

Process Understanding

IPCC AR4 FAQ,
extracted from UCAR The Climate System



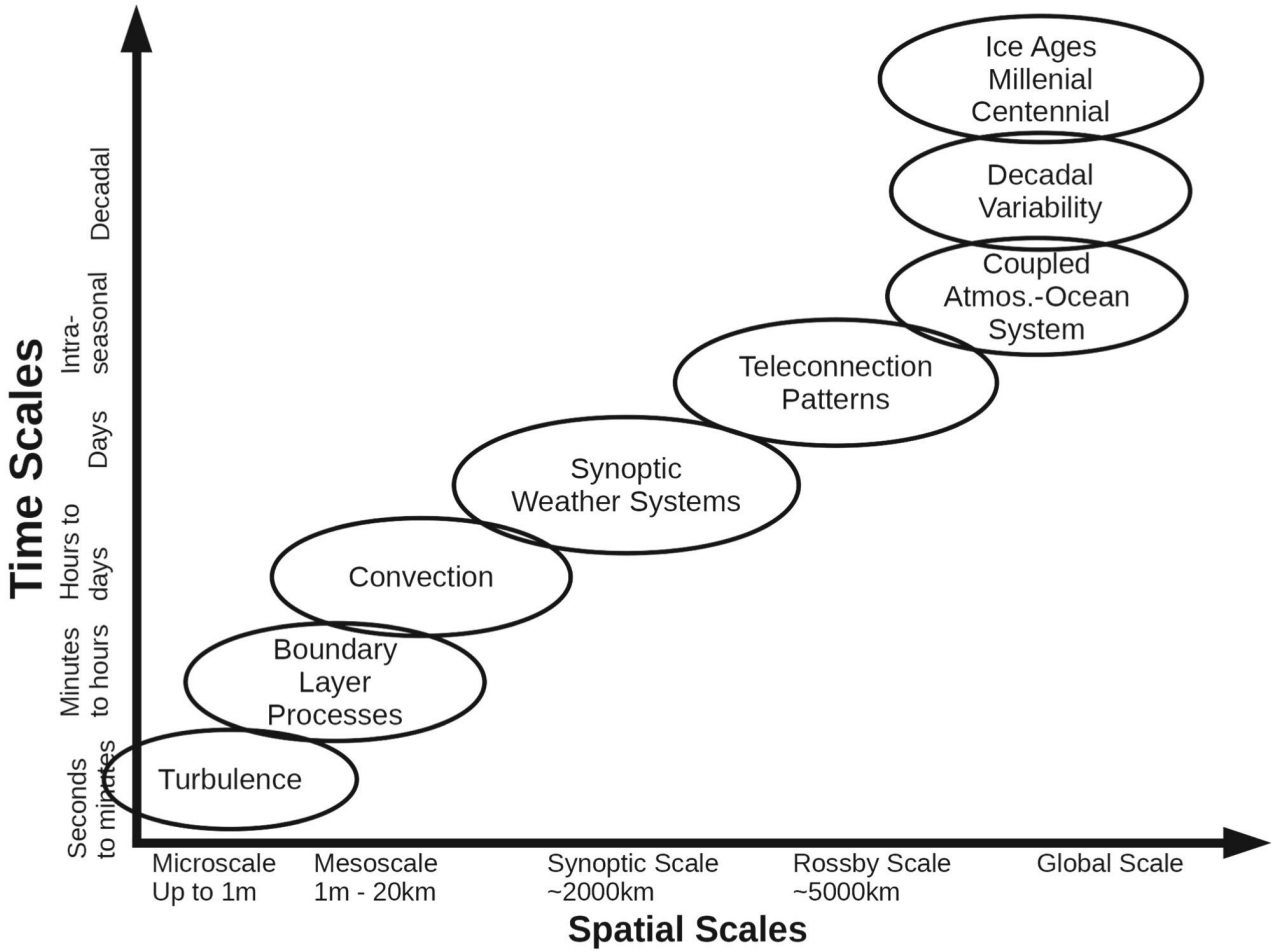

Weather Forecasting

NOAA National Weather Service



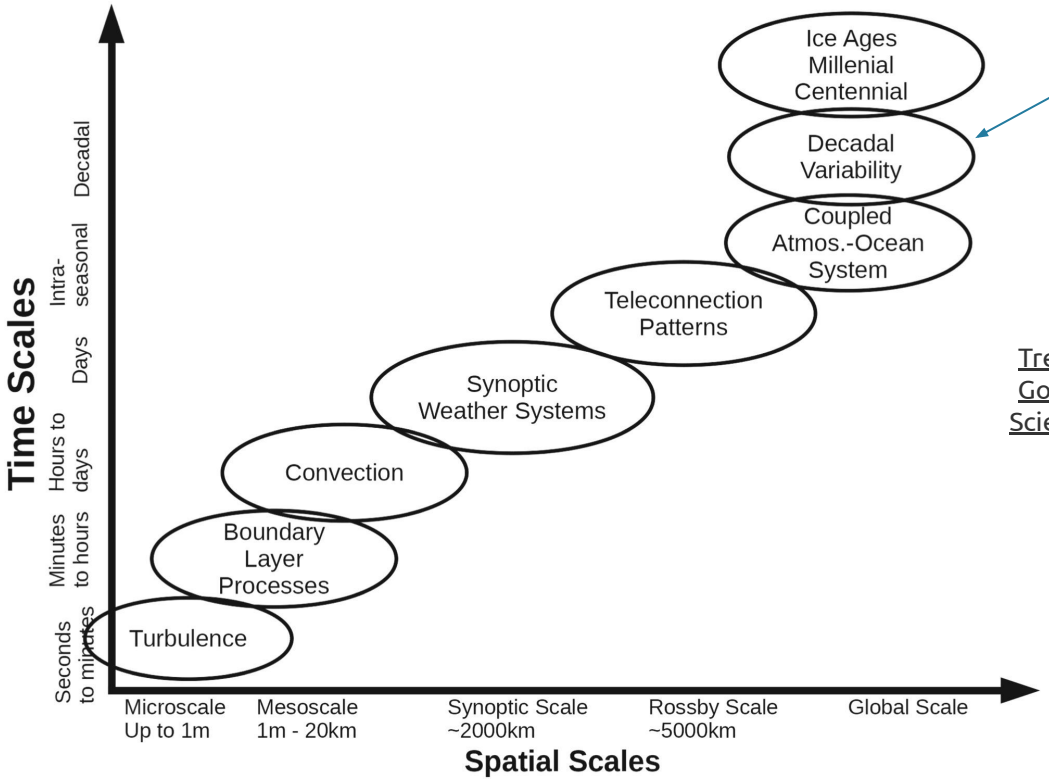

Numerical Models

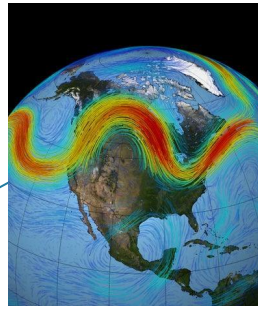
# Motivation: What are climate models?




Franzke et al. 2020

# Motivation: What are climate models



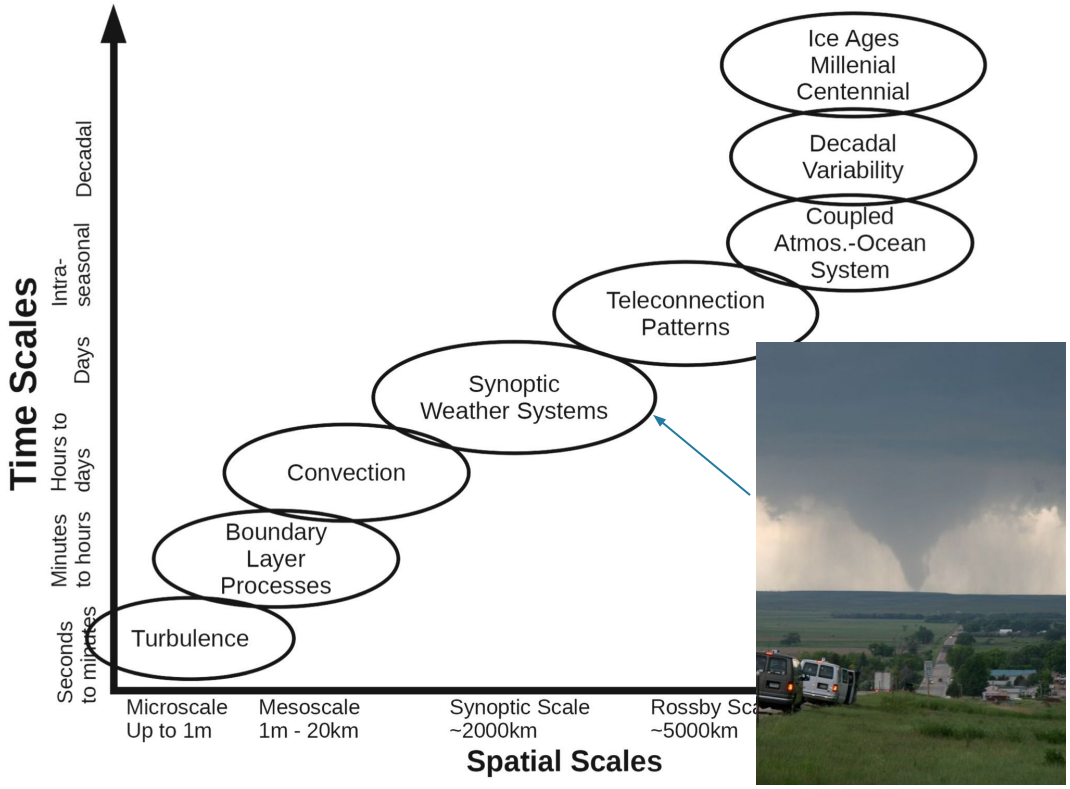



Jet Stream

$\sim 10^6$ m

Trent L. Schindler (NASA and Goddard Space Flight Center Scientific Visualization Studio)

# Motivation: Push towards Higher Resolution




Tornado
~$10^{3\text{-}4}$ m

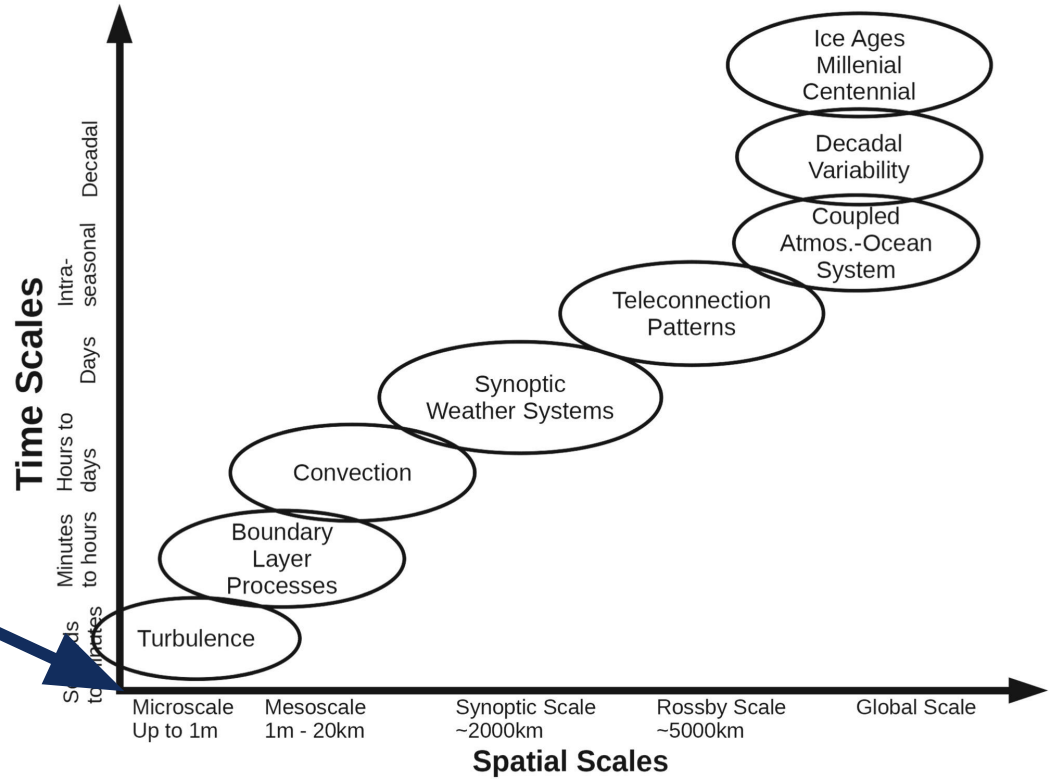
Bob Henson (NCAR/UCAR Image and Multimedia Gallery)

Franzke et al. 2020

# Motivation: What are climate models?

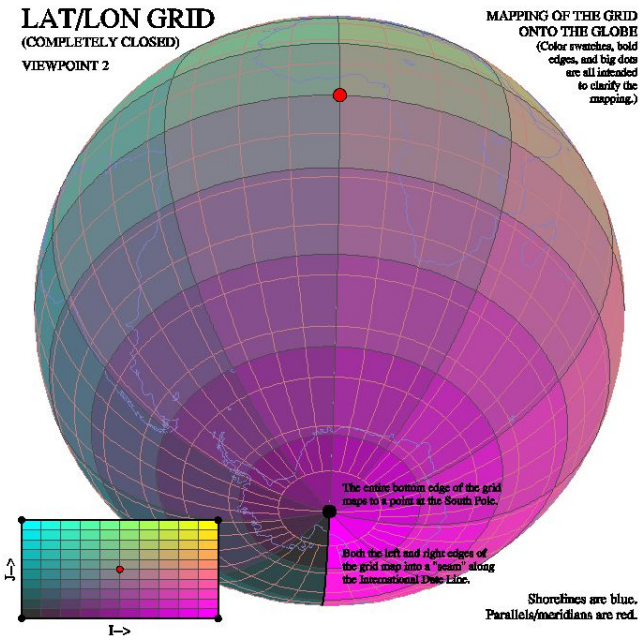
Snowflake
~$10^{-6}$ m

Reddit r/mildlyinteresting



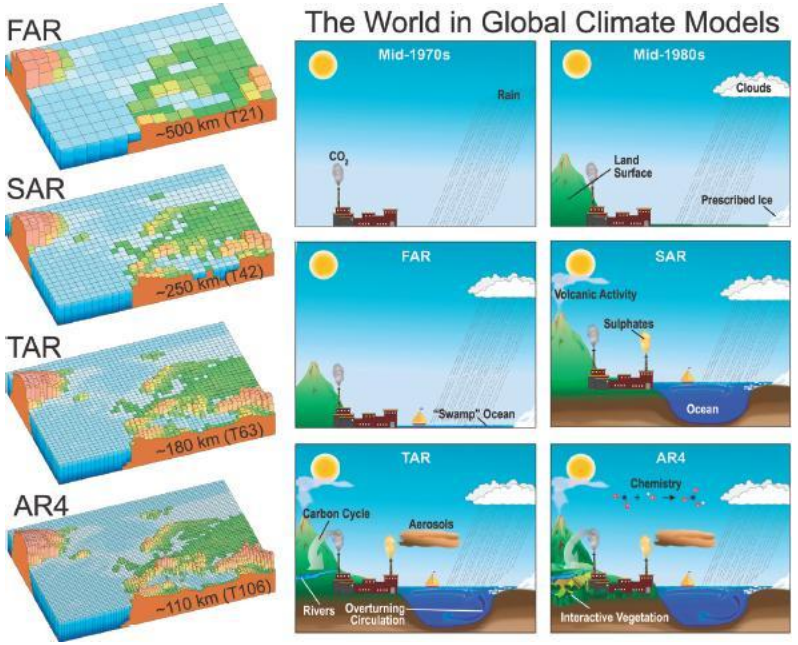

Franzke et al. 2020

# Motivation: Arrival of Kilometer-scale Global Models




Structured Lat-Lon Grid

NCAR Graphics

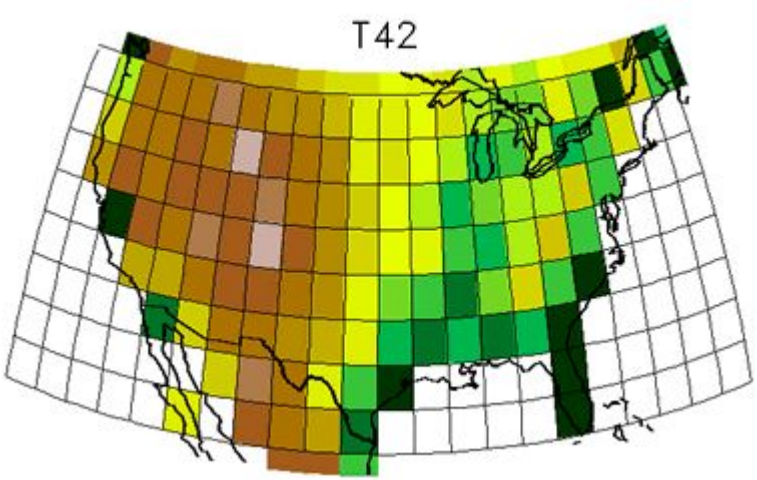



Increase in spatial resolution allows more climate process representations

Intergovernmental Panel on Climate Change (IPCC) AR4;
FAR/SAR/TAR: First/Second/Third Assessment Report, AR4: Fourth Assessment Report

**Limitation:** 2x Increase in Resolution requires 10x more computing power

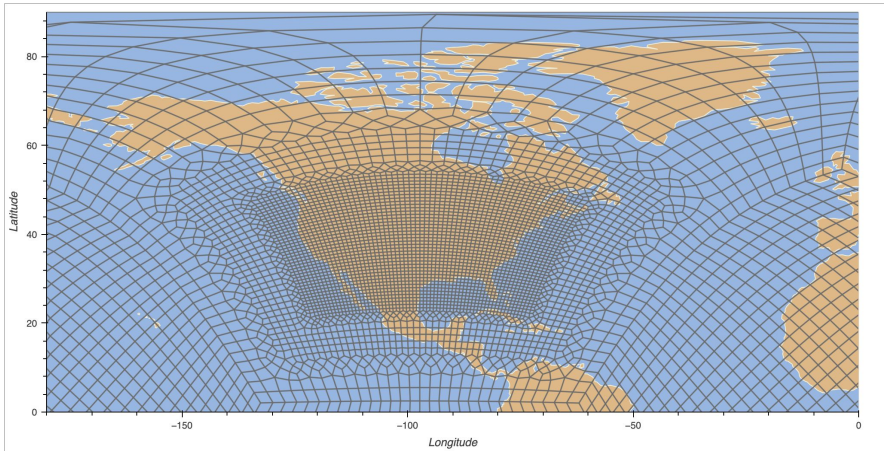
# Motivation: Arrival of Kilometer-scale Global Models




Higher resolution is required for answers to critical questions at regional levels.

Warren Washington/NCAR;
**T42:** mid-1990 typical model resolution; **T85:** 1 degree resolution (100 x 150km);
**T170 andT340:** High resolution

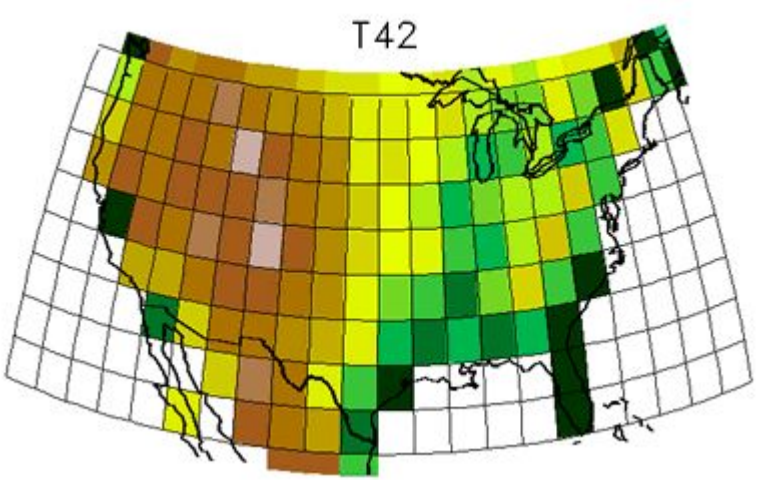
…thus drives motivation for transitioning to unstructured grids, which allows increase in resolution in certain regions.




Variable resolution, cubed sphere grid (CAM)

Project Raijin - UXarray

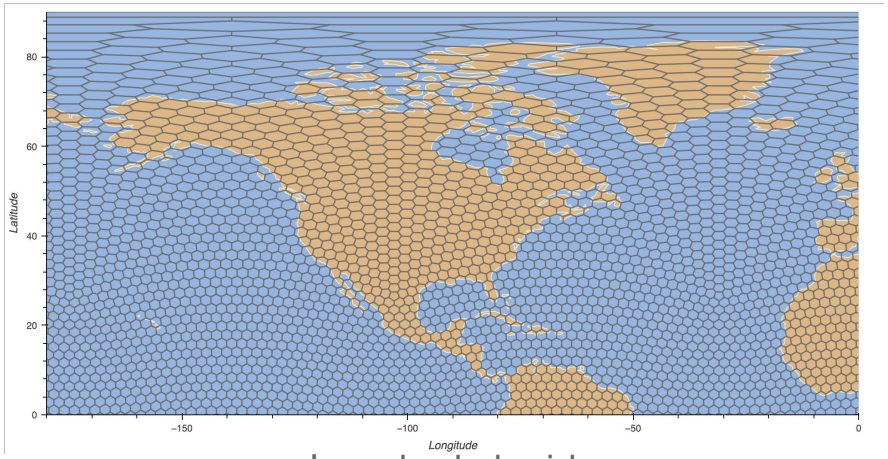
# Motivation: Arrival of Kilometer-scale Global Models




Higher resolution is required for answers to critical questions at regional levels.

Warren Washington/NCAR;
**T42:** mid-1990 typical model resolution; **T85:** 1 degree resolution (100 x 150km);
**T170 andT340:** High resolution

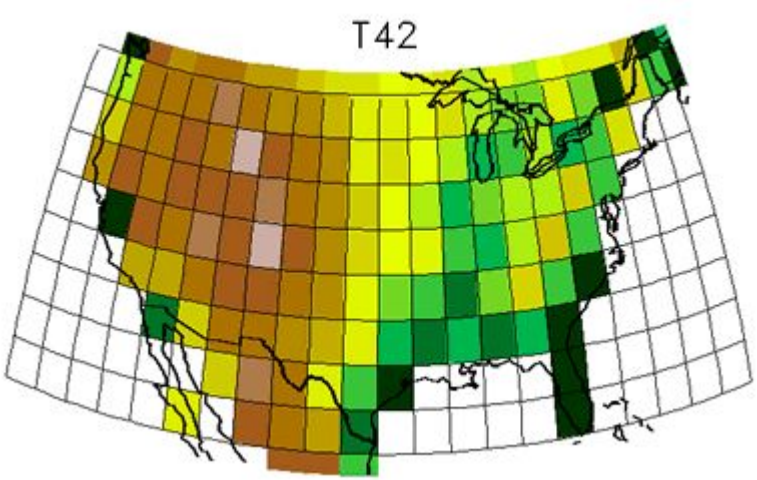
…thus drives motivation for transitioning to unstructured grids, which allows increase in resolution in certain regions.




Icosahedral grid

Project Raijin - UXarray

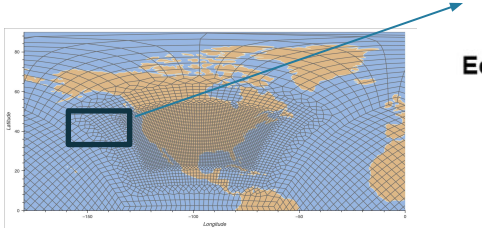
# Motivation: Arrival of Kilometer-scale Global Models

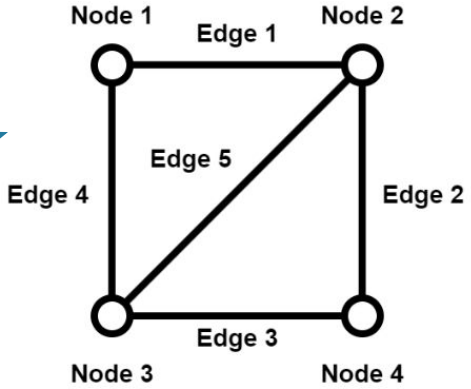



Higher resolution is required for answers to critical questions at regional levels.

Warren Washington/NCAR;
**T42:** mid-1990 typical model resolution; **T85:** 1 degree resolution (100 x 150km);
**T170 andT340:** High resolution

…thus drives motivation for transitioning to unstructured grids, which allows increase in resolution in certain regions.






Two triangular grid cells

Project Raijin UXarray

# Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation

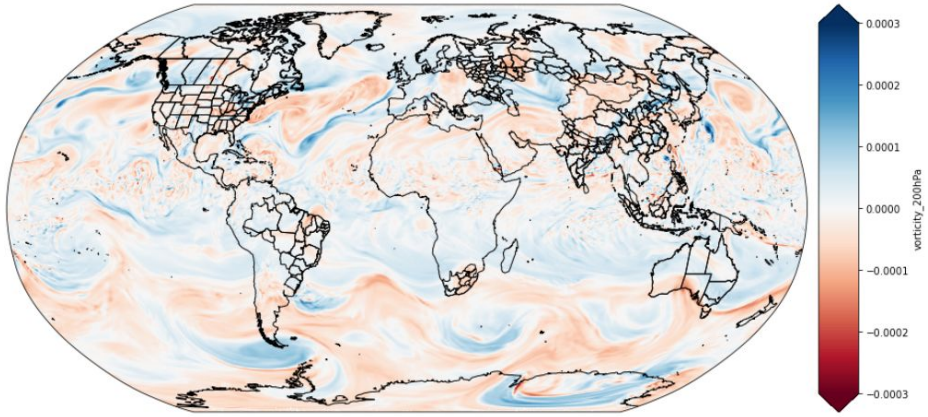

**Existing grid formats and files**

Details for each grid format available on Uxarray readthedocs

# Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation

2. Storm resolving resolutions generate a LOT of data




xarray.UxDataArray 'vorticity_200hPa' (**time**: 1, n_node: 83886080)

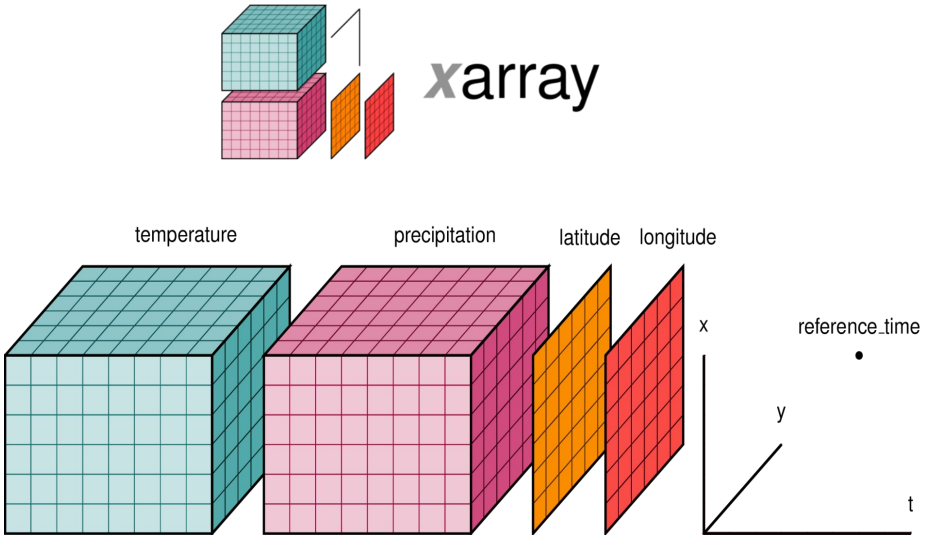
```
[83886080 values with dtype=float32]
```

Vorticity plotted from 3.75km MPAS Atmosphere mesh, composed of roughly 84 million nodes and 42 million faces at **a single timestep.**

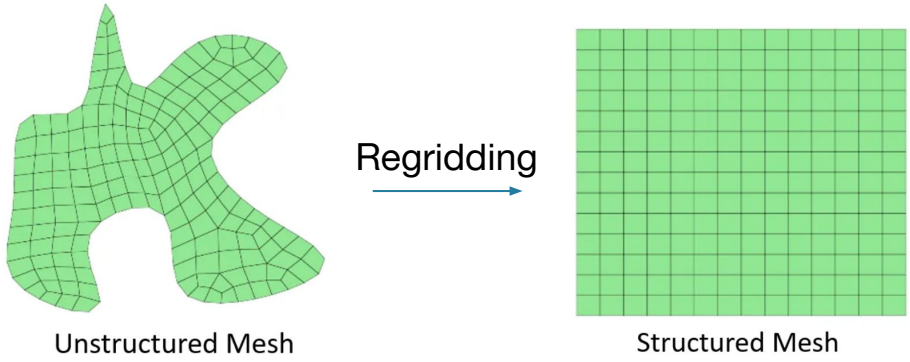
One file at this resolution in a single timestep has the size of **~10GB.**

# Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation
2. Storm resolving resolutions generate a LOT of data
3. Trivial structured grid operators typically need to be reimplemented




Xarray functionalities are built upon structured grids, which discrepancies are introduced when using unstructured grids.

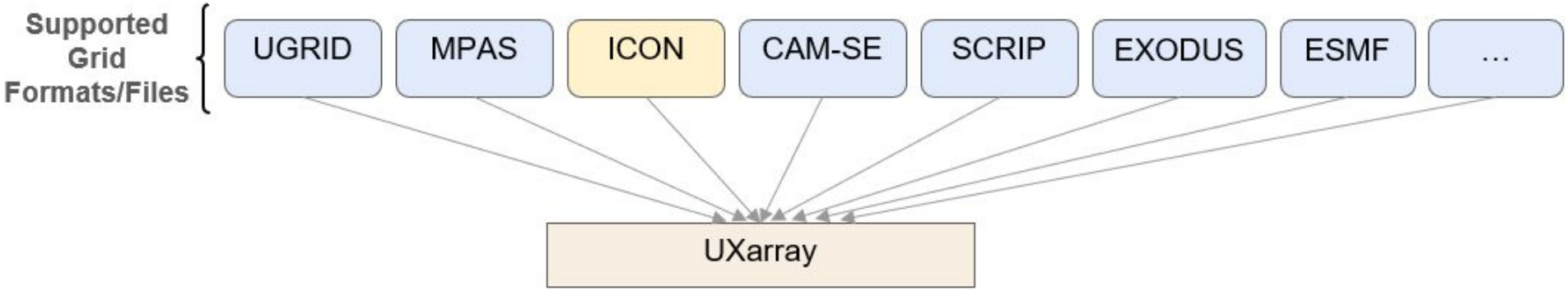


## Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation

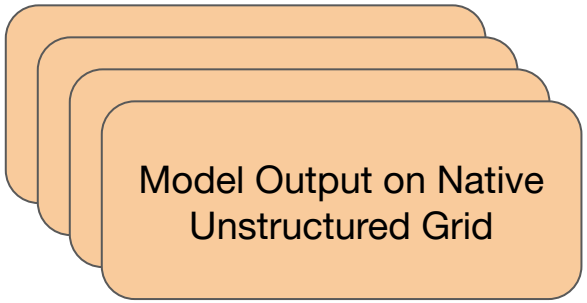
## Why Uxarray?

1. Supports wide range of unstructured grid conventions



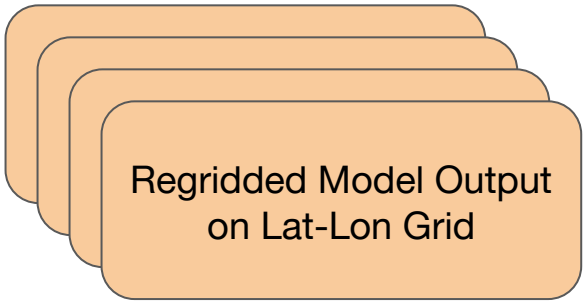
## Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation
2. Storm resolving resolutions generate a LOT of data




## Why Uxarray?

1. Supports wide range of unstructured grid conventions
2. Avoids duplication of memory consumption from regridding



## Challenges of Using Unstructured Grids

1. No widely-used convention for unstructured grid representation
2. Storm resolving resolutions generate a LOT of data
3. Trivial structured grid operators typically need to be reimplemented

## Why Uxarray?

1. Supports wide range of unstructured grid conventions
2. Avoids duplication of memory consumption from regridding
3. Avoids introduction of discrepancy from regridding

## Challenges of Using Unstructured Grids

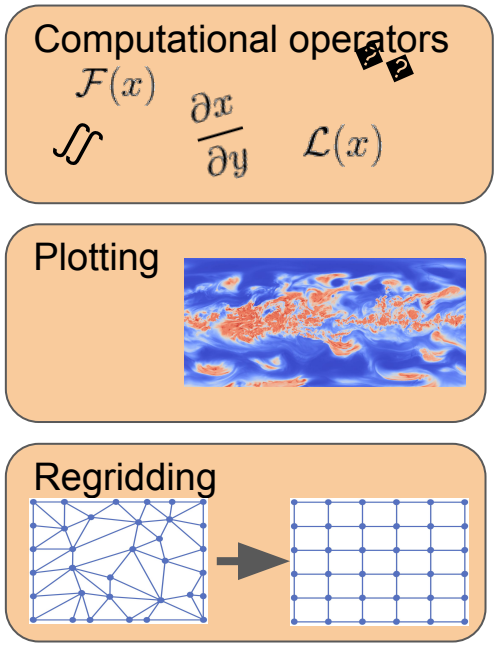
1. No widely-used convention for unstructured grid representation
2. Storm resolving resolutions generate a LOT of data
3. Trivial structured grid operators typically need to be reimplemented

## Why Uxarray?

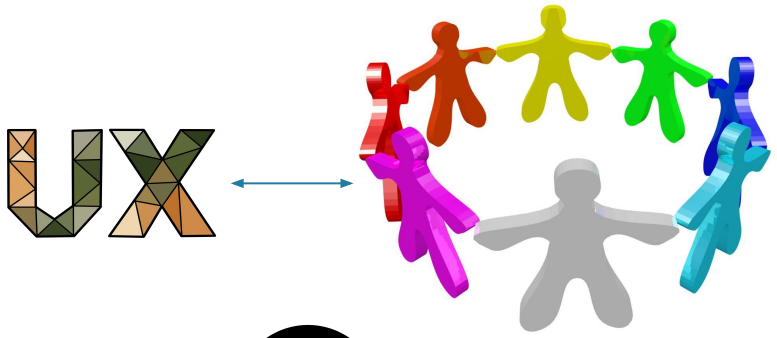
1. Supports wide range of unstructured grid conventions
2. Avoids duplication of memory consumption from regridding
3. Avoids introduction of discrepancy from regridding
4. Eliminates overhead from regridding

# Project Raijin and UXarray Goals

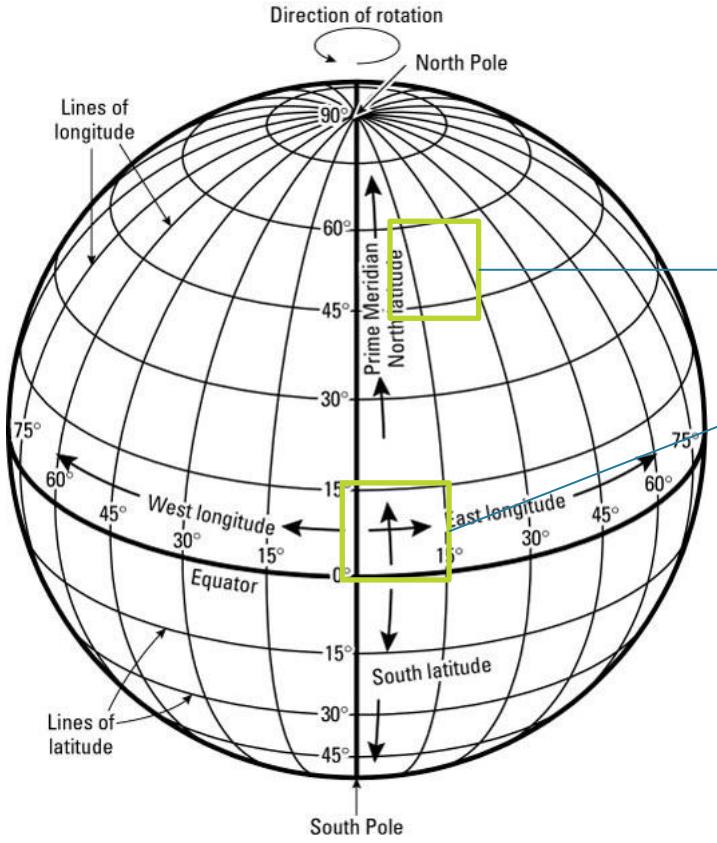
1. Develop extensible, scalable, open source software for data analysis on unstructured grids




2. Sustainable and community owned




# Routine Enhancement: Weighted Mean



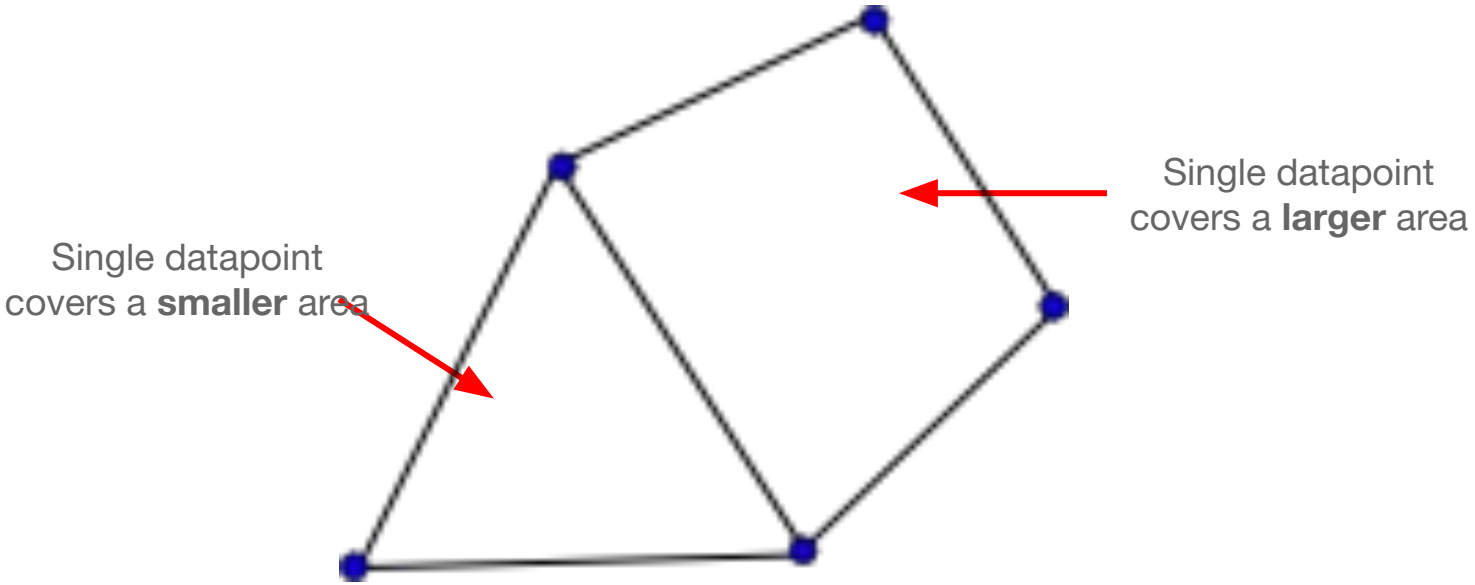

Grid box in the tropics would cover a spatial area that is larger than that in the midlatitudes.

Lat-lon grid

For Dummies

# Routine Enhancement: Weighted Mean

Regular mean would not take the area into account…




**…**Weighted mean will **normalize data by the weights**
(e.g. face area or edge length)

# Routine Enhancement: Weighted Mean

Large file size makes loading datasets in-memory impossible…

```
[2]: grid_path = "/glade/campaign/cisl/vast/uxarray/data/dyamond/3.75km/grid.nc"
     data_path = "/glade/campaign/mmm/wmr/fjudt/projects/dyamond_1/3.75km/diag.2016-08-01_00.00.00.nc"

[7]: uxds = ux.open_dataset(grid_path, data_path)
     uxds.nbytes

[7]: 17280533344
```

= 17.28 GB for a single timestep at 3.75km resolution

```
[8]: uxds

[8]: xarray.UxDataset
```

► Dimensions: (**time**: 1, StrLen: 64, n_face: 41943042, n_node: 83886080)

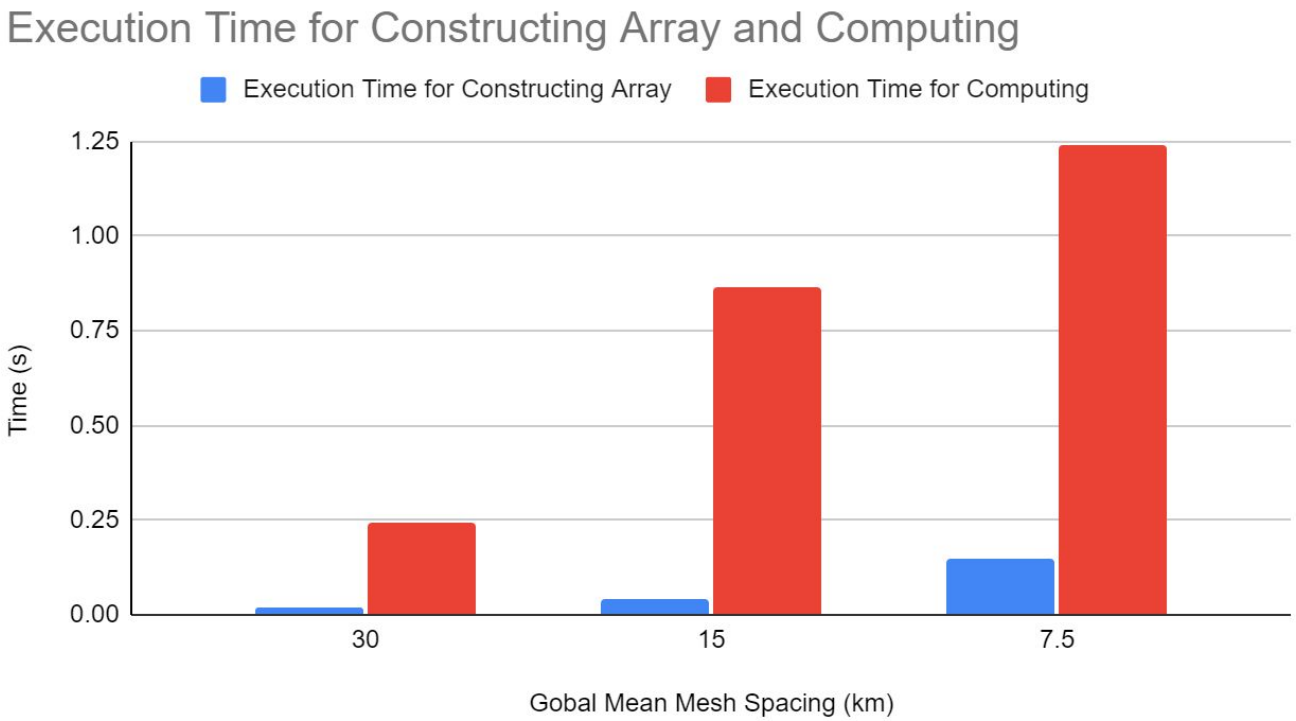
▼ Coordinates:

**time** (time) datetime64[ns] 2016-08-01

► Data variables: (99)

dask …but Dask implementations allows conducting operations in-parallel.

# Benchmarking for Weighted Mean




Data for benchmarking:
Synthetic Data on MPAS grid for DYAMOND, the global storm resolving models intercomparison project (Stevens et al., 2019)

# Documentation Enhancement for UXarray

## Loading Data with the `parallel` argument

Similar to Xarray, UXarray also supports loading data in parallel. Performance may not be significant due to the chosen dataset for this notebook; and Dask client configuration requires customization depending on the data.

The following code demonstrates setting up a local cluster with the use of 128 cores ( `n_workers` ), with 2 jobs ( `threads_per_worker` ) for each core.

```
%%time
# Regular Load
uxds_e3sm_basic_load = ux.open_mfdataset(grid_file, data_files, parallel=False)
```

```
CPU times: user 18.6 s, sys: 244 ms, total: 18.9 s
Wall time: 18.9 s
```

```
%%time
# Parallel Load
uxds_e3sm_parallel_load = ux.open_mfdataset(grid_file, data_files, parallel=True)
```

```
CPU times: user 11.5 s, sys: 1.5 s, total: 13 s
Wall time: 12.7 s
```

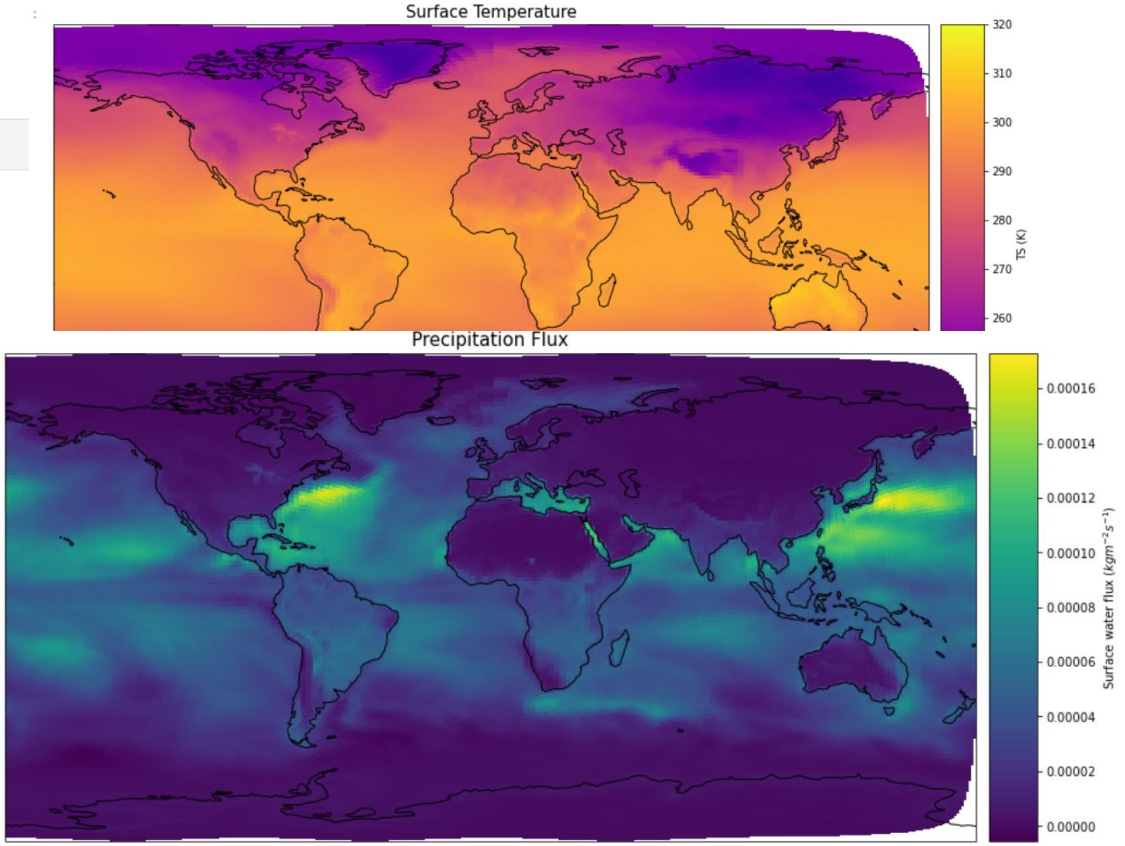
Populate user guide for Uxarray – **Dask Implementation for Parallel Loading**

# Documentation Enhancement for UXarray

```
uxds_e3sm_multi
```

```
<xarray.UxDataset>
Dimensions:                (time: 72, n_face: 21600, lev: 72, ilev: 73,
                            cosp_prs: 7, nbnd: 2, cosp_tau: 7, cosp_scol: 10,
                            cosp_ht: 40, cosp_sr: 15, cosp_sza: 5,
                            cosp_htmisr: 16, cosp_tau_modis: 7, cosp_reffice: 6,
                            cosp_reffliq: 6)
Coordinates: (12/13)
  * lev                    (lev) float64 0.1238 0.1828 0.2699 ... 993.8 998.5
  * ilev                   (ilev) float64 0.1 0.1477 0.218 ... 990.5 997.0 1e+03
  * cosp_prs               (cosp_prs) float64 9e+04 7.4e+04 ... 2.45e+04 9e+03
  * cosp_tau               (cosp_tau) float64 0.15 0.8 2.45 6.5 16.2 41.5 100.0
  * cosp_scol              (cosp_scol) int32 1 2 3 4 5 6 7 8 9 10
  * cosp_ht                (cosp_ht) float64 1.896e+04 1.848e+04 ... 720.0 240.0
    ...                     ...
  * cosp_sza               (cosp_sza) float64 0.0 20.0 40.0 60.0 80.0
  * cosp_htmisr            (cosp_htmisr) float64 0.0 250.0 ... 1.6e+04 1.8e+04
  * cosp_tau_modis         (cosp_tau_modis) float64 0.15 0.8 2.45 ... 41.5 100.0
  * cosp_reffice           (cosp_reffice) float64 5e-06 1.5e-05 ... 5e-05 7.5e-05
  * cosp_reffliq           (cosp_reffliq) float64 4e-06 9e-06 ... 1.75e-05 2.5e-05
  * time                   (time) object 0001-02-01 00:00:00 ... 0007-01-01 00:...
Dimensions without coordinates: n_face, nbnd
Data variables: (12/471)
```

```
CPU times: user 4.48 s, sys: 559 ms, total: 5.04 s
Wall time: 4.98 s
```




Populate UXarray usage examples – **Visualization with Holoviz with E3SMv2 EAM 1-degree resolution**

# Documentation Enhancement for UXarray

### Calculating Cloud Radiative Effect (netCRE) with UXarray

Cloud radiative effect can be calculated as follows:

Shortwave cloud radiative effect can be approximated as the difference beteween all-sky net shortwave flux (FSNT) at the top of the model and the clear-sky net shortwave flux (FSNTC).

```
{math}
        SWCRE = FSNT - FSNTC
```

Longwave cloud radiative effect is similar to that for SWCRE, but the all-sky and clear-sky longwave fluxes are applied instead.

```
{math}
    LWCRE = FLUT - FLUT
```

Net cloud radiative effect is thus the difference beteween shortwave and longwave cloud radiative effect.

```
{math}
    netCRE = SWCRE - LWCRE
```

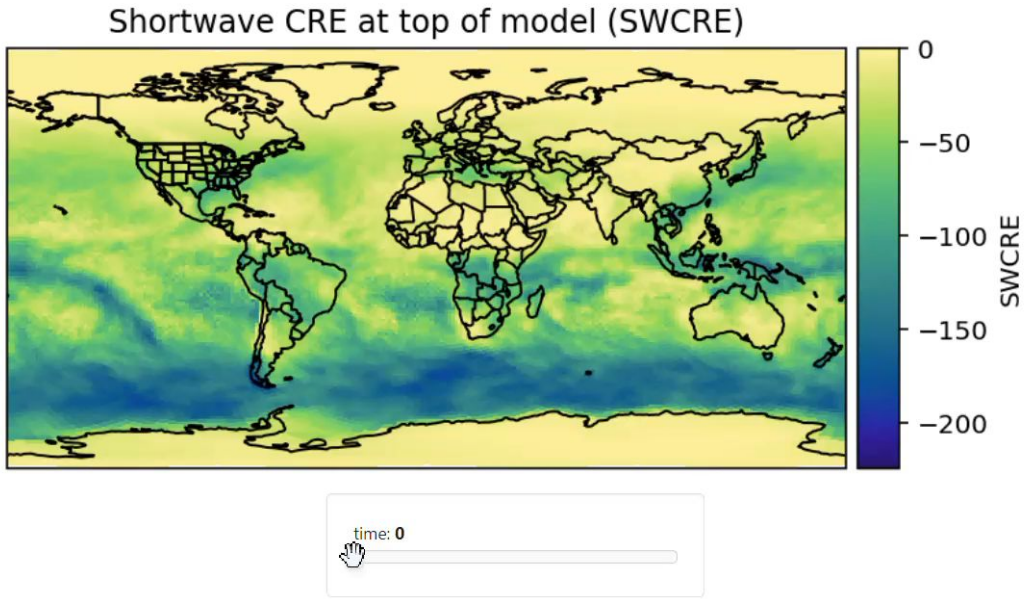
### Calculate Shortwave Cloud Radiative Effect (SWCRE)

All of the following dataarrays are lazily loaded, as demonstrated below.

```
[5]:
# Define function to display variable size in MiB
def var_size(in_var):
    result = sys.getsizeof(in_var) / 1024 / 1024
    return result


# Loading Variables
FSNT = uxds_e3sm_multi["FSNT"]
FSNTC = uxds_e3sm_multi["FSNTC"]

SWCRE = FSNT - FSNTC
SWCRE.name = "SWCRE"  # UXarray requires dataarray to have a name
```




Populate UXarray usage examples – **Conducting operations with UXarray and Visualization with Holoviz**

# Acknowledgements

Google profile photo

Google profile photo

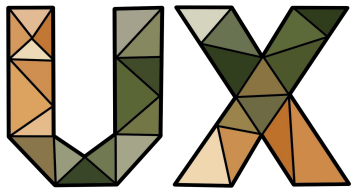

GeoCAT

PROJECT raijin

SIParCS Summer Internships in Parallel Computational Science BOULDER • CO

# Summary and Future Steps

1. UXarray meets the need of unstructured-grid data analysis
2. Uxarray is now progressing towards higher scalabilty, in parallel to populating with more functionalities on unstructured grids.
3. UXarray always welcome user feedback for improvements.



User Guide

GitHub