High Performance Research Computing
DIVISION OF RESEARCH

TEXAS A&M UNIVERSITY
Oceanography

Post-processing CESM model output using Python-based utilities

Fall 2020 HPRC Short Course
Oct 12, 2020

Abishek Gopal
Assistant Research Scientist
iHESP, Texas A&M Oceanography
Texas A&M High Performance Research Computing

Helpful HPRC resources

- Ada quick start guide
  - https://hprc.tamu.edu/wiki/Ada:QuickStart
- Introduction to the Ada and Terra clusters – Short course (Spring 2020)
  - https://hprc.tamu.edu/files/training/2020/Spring/Intro_to_HPRC_clusters_2020_spring.pdf
- Submit tickets to help@hprc.tamu.edu

Upcoming relevant HPRC short courses

- Oct 23: Introduction to Python
  - **Instructor:** Keith Jackson
  - **Time:** Friday, October 23, 10:00AM - 12:30PM
- Oct 30: Introduction to Scientific Python
  - **Instructor:** Tri Minh Pham
  - **Time:** Friday, October 30, 10:00AM - 12:30PM
- Oct 30: AI techniques usage – Jupyter Notebook
  - **Instructor:** Jian Tao
  - **Time:** Friday, October 30, 1:30PM - 4:00PM

https://hprc.tamu.edu/training/index.html

Course structure

- Structure of CESM model output and netCDF format
- Installing and configuring the CESM_postprocessing package
- Generating time-averages and climatology files
- Short break (15 minutes)
- Generating diagnostics of CESM model runs
- Pangeo for geosciences
- Loading and visualizing datasets using xarray and cartopy
- Speeding up computations using dask

Target audience

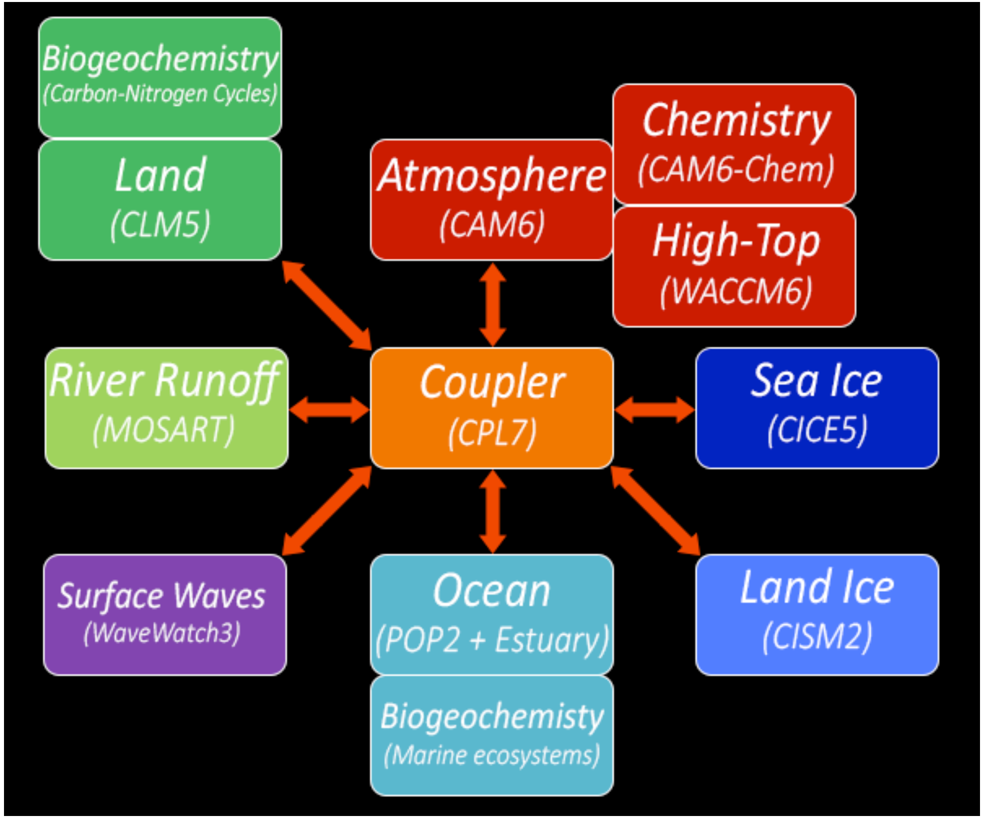
You're a geoscience researcher who

- Performs your own CESM model runs
- Analyzes publicly-available CESM model data
- Uses serial NCO,NCL,CDO,MATLAB or Ferret programs for analysis
- Works with other GCMs

Expectations for this course

- Get an overview of the existing parallel python tools to post-process climate model output
- Learn about the available diagnostic plots sets
- Obtain some hands-on experience with the CESM postprocessing tool
- Get a peek at some of the upcoming trends in using Python for geosciences

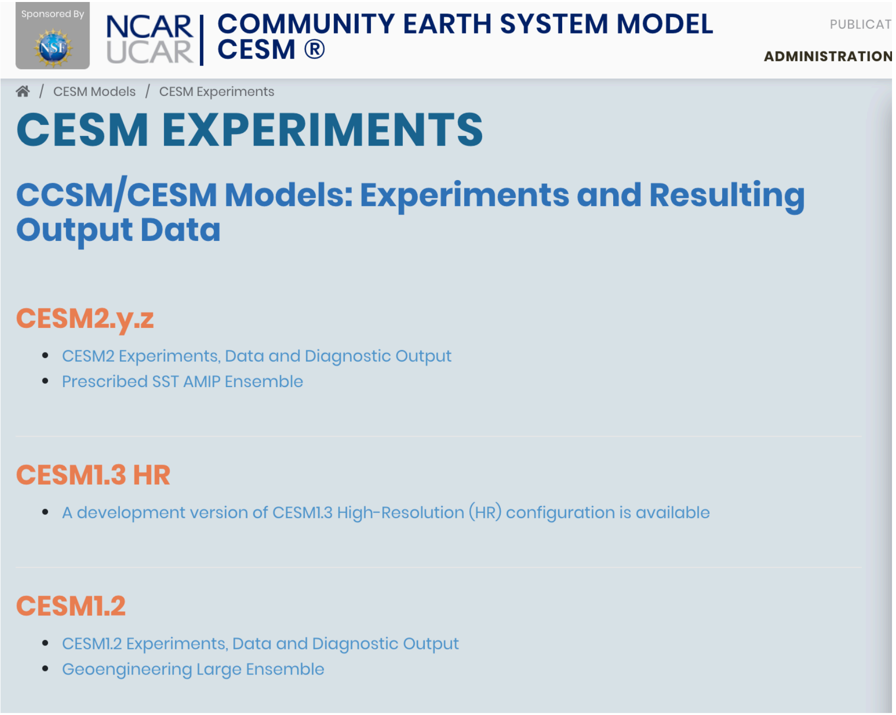
The Community Earth System Model (CESM)



https://nar.ucar.edu/2017/cgd/development-cesm2

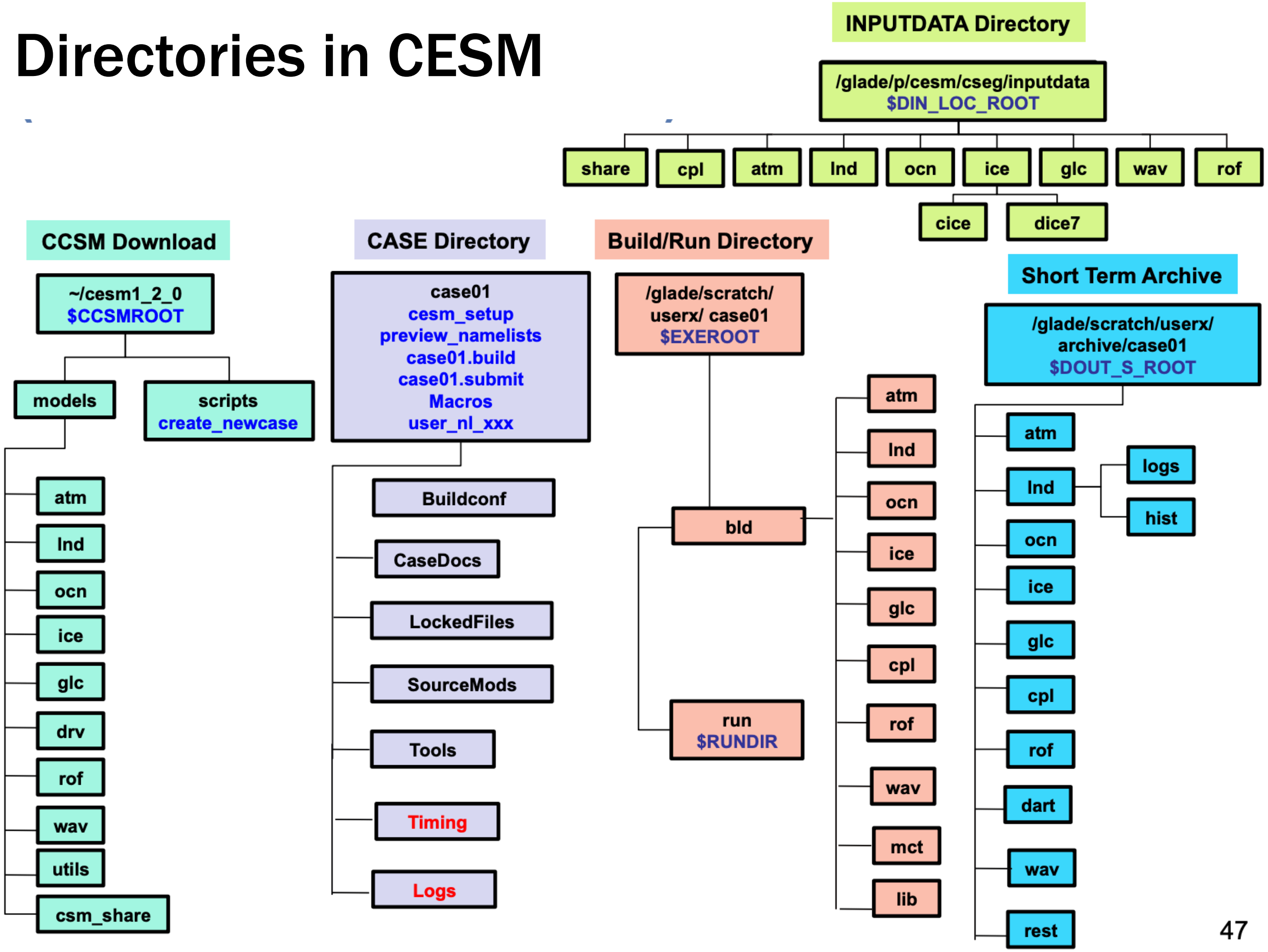
Publicly available CESM model output

- http://www.cesm.ucar.edu/experiments/
- https://www.earthsystemgrid.org/



Sample datasets for the tutorial available on Ada at /scratch/training/CESM_post/datasets

Directories in CESM



http://www.cesm.ucar.edu/models/cesm1.2/cesm/cesm1_tutorial_130619_full.pdf

History vs Time series format

- History files contain a set of variables output at a given time slice in a single file.
- Time series files contain single variable data for all times
- CESM writes model output in short-term archive as history files
- Variables with larger dimensions (Ex: TEMP) are usually output as a lower frequency history files, than smaller variables (Ex: SST).
- Time series files are usually shared publicly.
- How to transpose history files to time series files?

CESM Case naming convention

<compset char>.<code base>.<compset sname>.<resolution sname>[.opt_desc_string].<nnn>[opt_char]

<compset char>	= 1 character, first letter of compset
<code base>	= code base, "e20" for cesm2.0, "e10" for cesm1.0, "c40" for ccsm4.0, and "c35" for ccsm3.5.
<compset sname>	= compset shortname
<resolution sname>	= resolution shortname
<.opt_desc_string>	= optional descriptive string, to be kept short if possible
<nnn>	= 3 digit number
[opt_char]	= optional single lower-cased letter; allowed to distinguish a group of cases that are very closely related

Examples: b.e21.B1850.f09_g16.CMIP6-piControl.001
f.e11.FAMIPCN.f09_f09.rcp85_ersstv5.005

One note about compsets: The first letter of the casename is indicative of the type of run it is: A coupled run (A "B" case), an atmosphere/land run ("F"), a land run ("I"), or an ocean/ice run ("G"). Other letters (A,Q,S,T,X) denoting less common configurations are also used.

Adam Phillips, NCAR

NetCDF format

```
netcdf pres_temp_4D {
  dimensions:
                  level = 2 ;
                  latitude = 6 ;
                  longitude = 12 ;
                  time = UNLIMITED ;
  variables:
                  float latitude(latitude);
                           latitude:units = "degrees_north" ;
                  float longitude(longitude) ;
                           longitude:units = "degrees_east" ;
                  float pressure(time, level, latitude, longitude) ;
                           pressure:units = "hPa" ;
                  float temperature(time, level, latitude, longitude) ;
                                              temperature:units = "celsius" ;
  data:
                  latitude = 25, 30, 35, 40, 45, 50 ;
                  longitude = -125, -120, … ;
                  pressure = 900, 901, 902, … ;
                  temperature = 9, 10, 11, …;
}
```

https://www.archer.ac.uk/training/course-material/2015/01/data_mgmt_epcc/netcdf.pdf

Tools to process and visualize NetCDF files

- NetCDF Operators (NCO)
- Climate Data Operators (CDO)
- NCAR Command Language (NCL)
- MATLAB
- Interactive Data Language (IDL)
- Panoply (from NASA GISS)
- PyNGL and PyNIO

Software for Manipulating or Displaying NetCDF Data

This document provides references to software packages that may be used for manipulatin freely-available and licensed (commercial) software that can be used with netCDF data. We corrections or additions, please send them to us. Where practical, we would like to include version of this document.

Other useful guides to utilities that can handle netCDF data include ARM's list of ARM-test and the NOAA Geophysical Fluid Dynamics Laboratory guide to netCDF utilities.

Freely Available Software

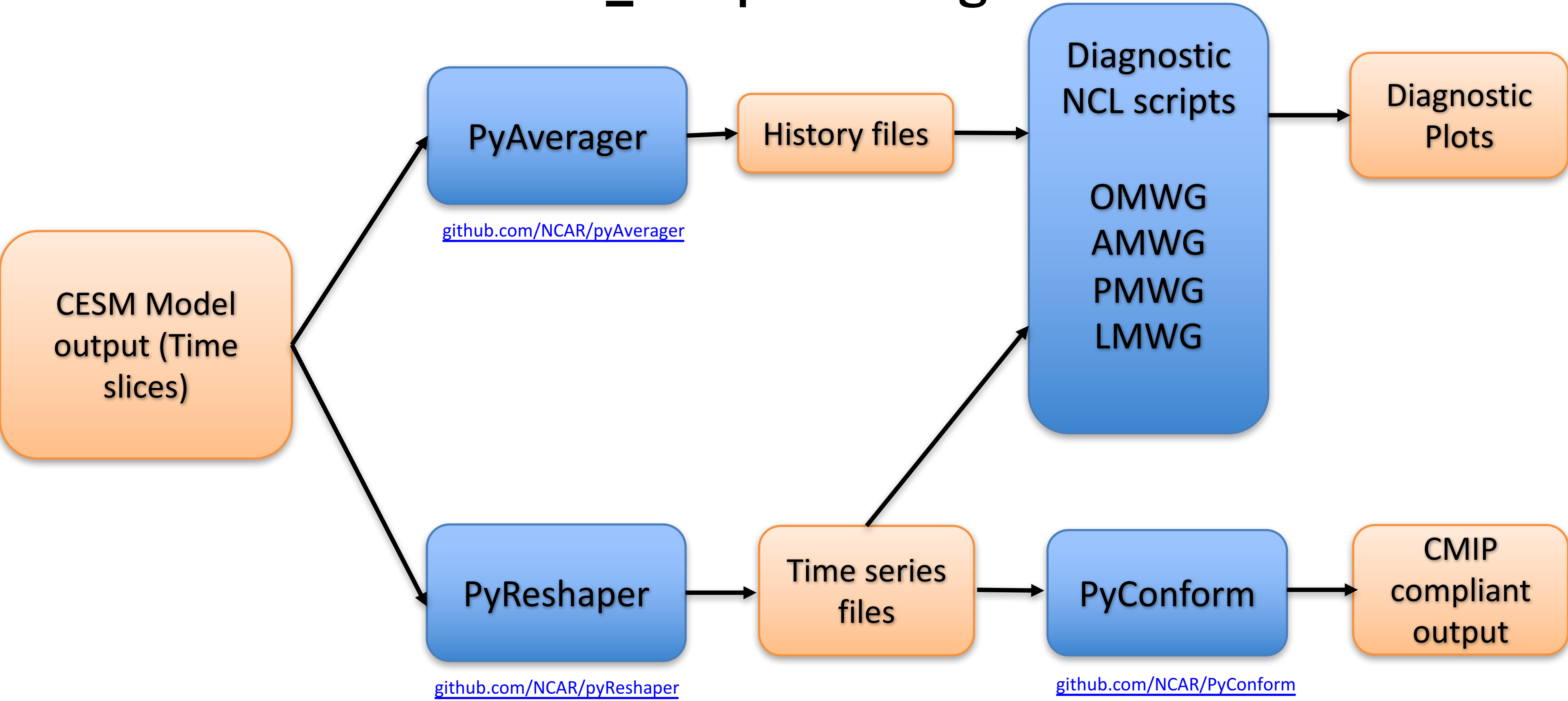
- ANDX (ARM NetCDF Data eXtract) and ANAX (ARM NetCDF ASCII eXtract)
- ANTS (ARM NetCDF Tool Suite)
- ARGOS (interActive thRee-dimensional Graphics ObServatory)
- CDAT (Climate Data Analysis Tool)
- CDFconvert (Convert netCDF to RPN and GEMPAK Grids)
- cdfsync (network synchronization of netCDF files)
- CDO (Climate Data Operators)
- CIDS Tools
- CSIRO MATLAB/netCDF interface
- EPIC
- Excel Use
- EzGet
- FAN (File Array Notation)
- FERRET
- FIMEX (File Interpolation, Manipulation, and EXtraction)
- FWTools (GIS Binary Kit for Windows and Linux)
- GDAL (Geospatial Data Abstraction Library)
- GDL (GNU Data Language)
- Gfdnavi (Geophysical fluid data navigator)
- Gliderscope
- GMT (Generic Mapping Tools)
- Grace
- GrADS (Grid Analysis and Display System)
- Gri
- GXSM - Gnome X Scanning Microscopy project
- HDF (Hierarchical Data Format) interface
- HDF-EOS to netCDF converter
- HIPHOP (Handy IDL-Program for HDF-Output Plotting)

https://www.unidata.ucar.edu/software/netcdf/software.html

Typical post-processing workflow

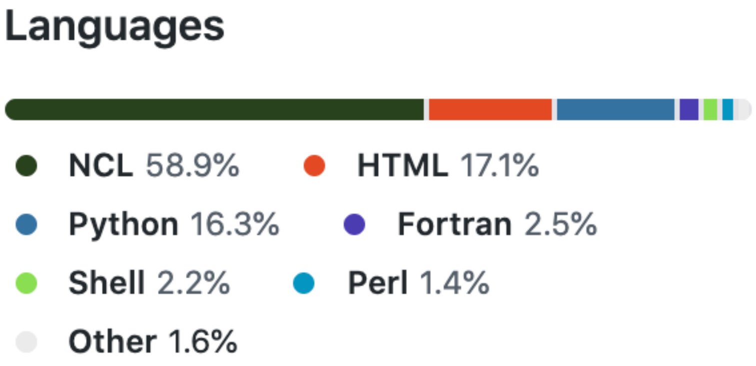
1. Convert history to time series files (optional)
2. Interpolate quantities from native grid onto a regular grid (optional)
3. Compute time average, climatology, zonal averages, etc
4. Generate diagnostics plots from derived fields
5. Convert time series to CMIP6-compliant output (optional)

Structure of the CESM_Postprocessing tool



Structure of the CESM_Postprocessing tool

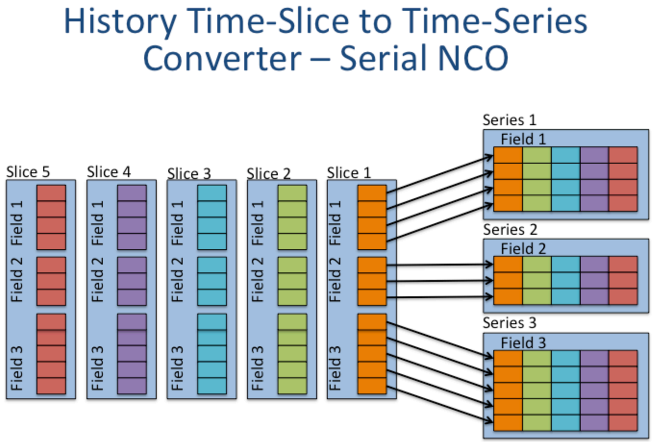
- Python scripts parallelized with MPI for the intensive data processing steps
  - Averaging
  - Transposition
  - CMORization
  - Regridding

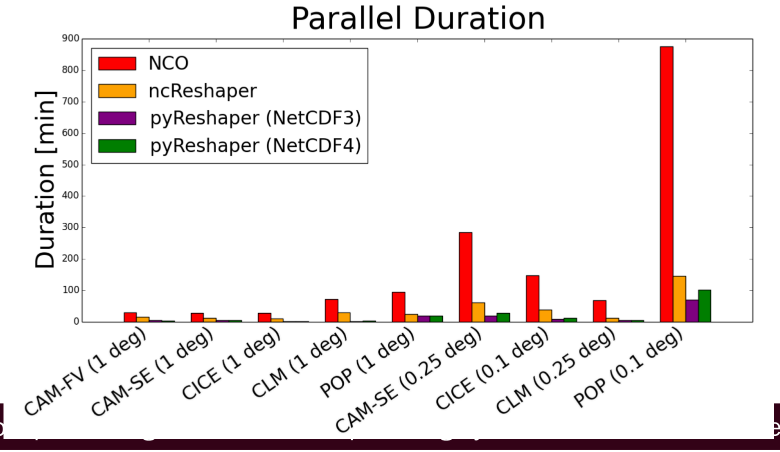


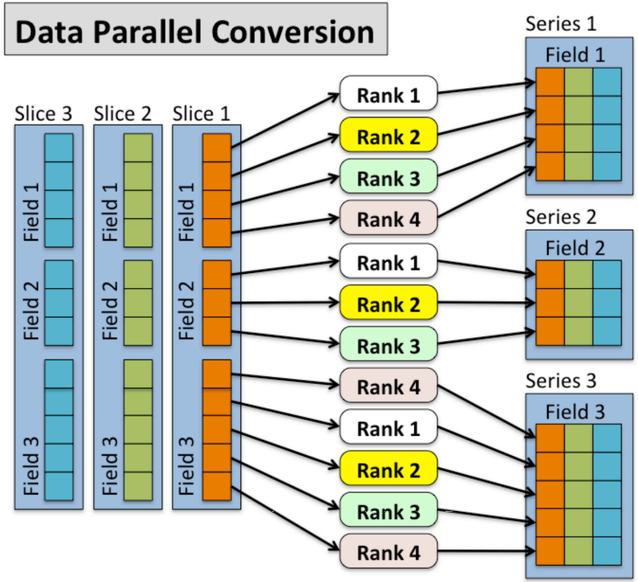
https://github.com/NCAR/CESM_postprocessing

- NCL for regridding, generating diagnostic plots, etc

pyReshaper vs NCO

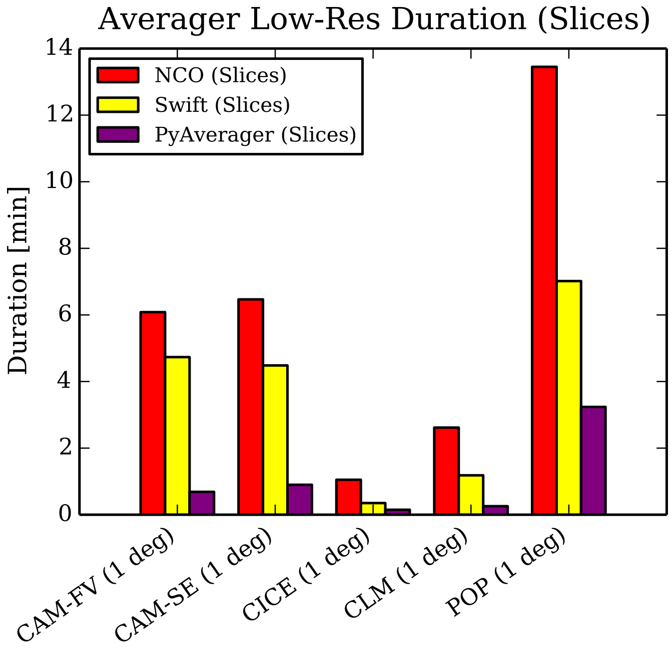


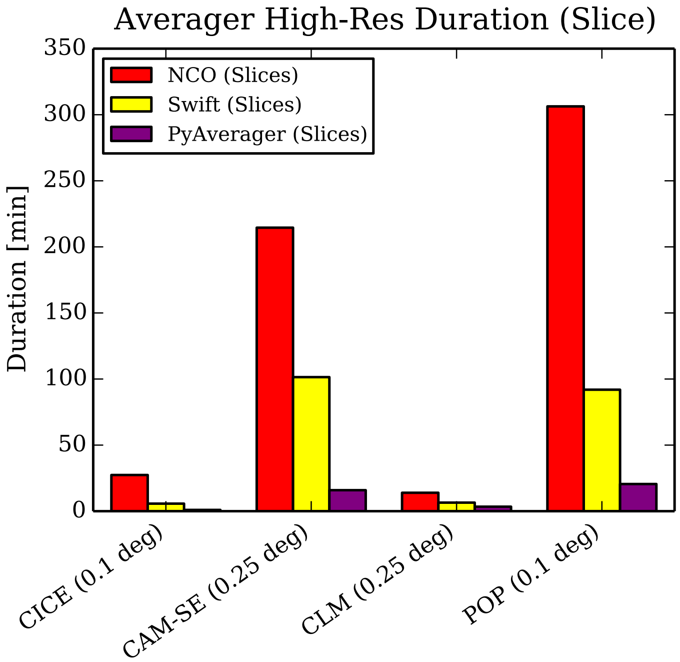




Paul K, Dennis M, Xu H, Mickelson S, Vertenstein M, Bertini A, Edwards J. PyReshaper: A parallel Python tool for NetCDF time-slice to time-series conversion [poster].

PyAverager vs NCO





Sheri Mickelson[1] & Alice Bertini[2, NCAR]

POP 0.1deg grid = 15 times faster than NCO

Generated diagnostics

AMWG Diagnostics Package

b.e21.B1850.f09_g17.CMIP6-piControl.001 and b.e21.B1850.f09_g17.CMIP6-piControl.001

NCAR
National Center for Atmospheric Research
Boulder, Colorado

Plots Created
Tue Aug 28 11:32:34 2018

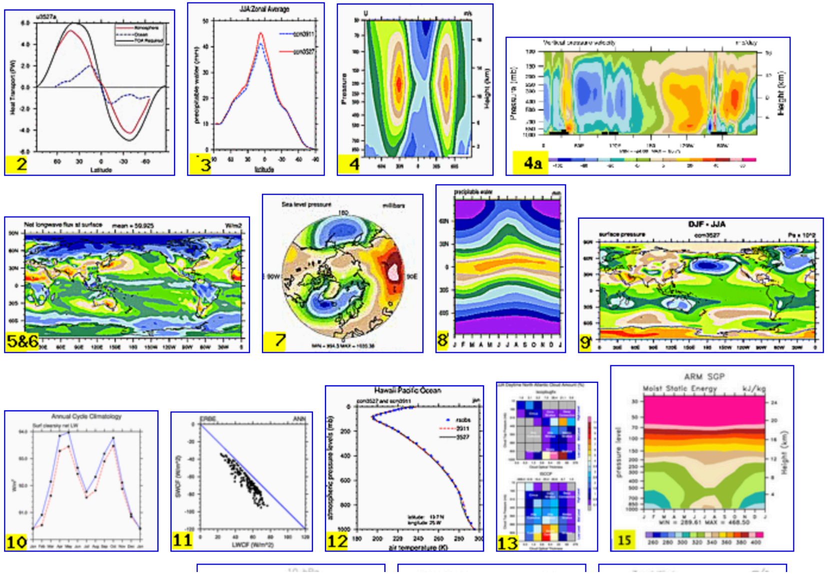
Set Description

1 Tables of ANN, DJF, MAM, JJA, SON, global and regional means and RMSE.
2 Line plots of annual implied northward transports.
3 Line plots of DJF, MAM, JJA, SON and ANN zonal means
4 Vertical contour plots of DJF, MAM, JJA, SON and ANN zonal means
4a Vertical (XZ) contour plots of DJF, MAM, JJA, SON and ANN meridional means
5 Horizontal contour plots of DJF, MAM, JJA, SON and ANN means
6 Horizontal vector plots of DJF, MAM, JJA, SON and ANN means
7 Polar contour and vector plots of DJF, MAM, JJA, SON and ANN means
8 Annual cycle contour plots of zonal means
9 Horizontal contour plots of DJF-JJA differences
10 Annual cycle line plots of global means
11 Pacific annual cycle, Scatter plot plots
12 Vertical profile plots from 17 selected stations
13 Cloud simulator plots
14 Taylor Diagram plots
15 Annual Cycles at Select Stations plots
16 Budget Terms at Select Locations plots

WACCM Set Description

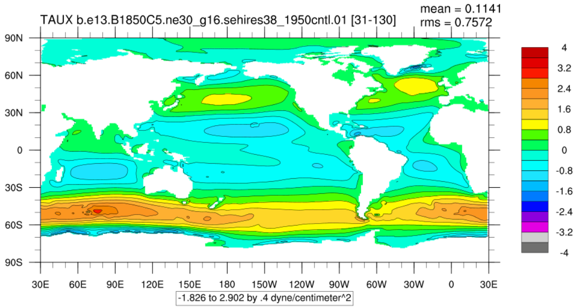
1 Tables of regional min, max, means

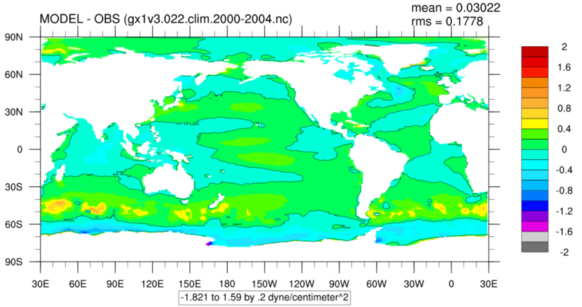
Click on Plot Type



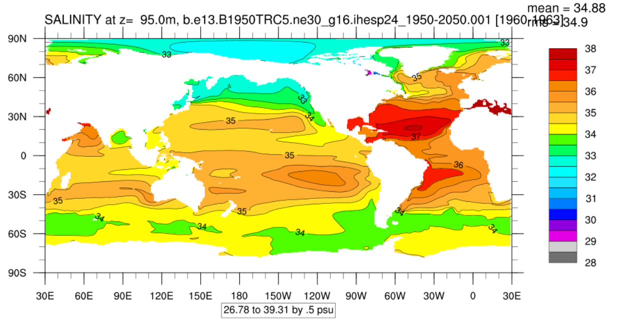
Categories of diagnostic sets

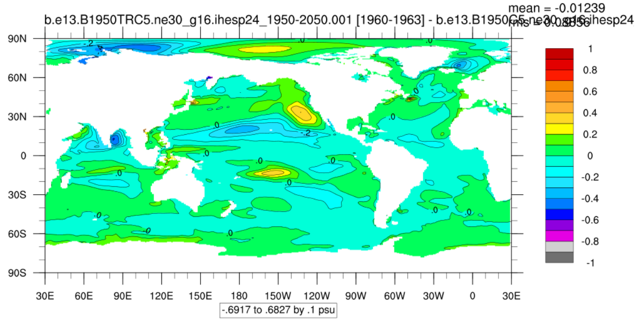
Model vs Observations



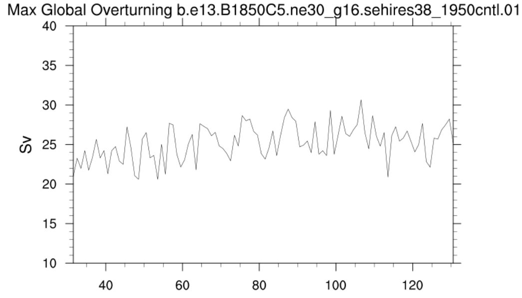


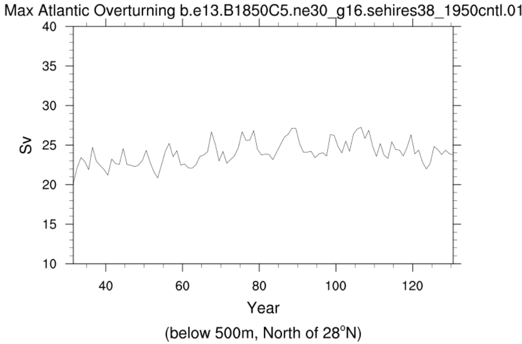
Model vs Control





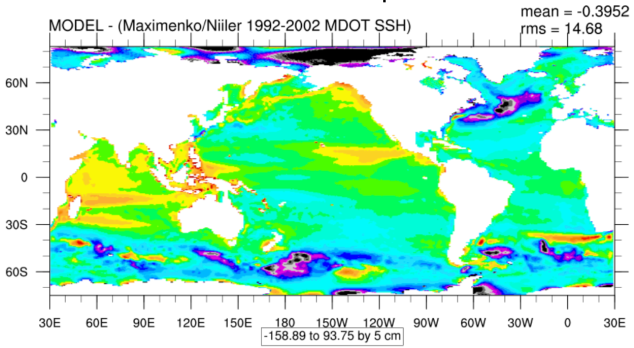
Model time series



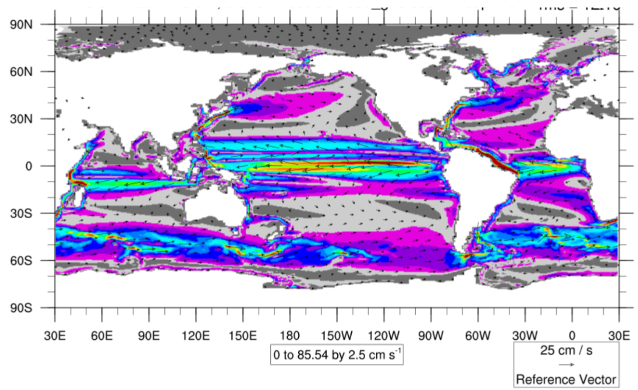


Ocean diagnostics

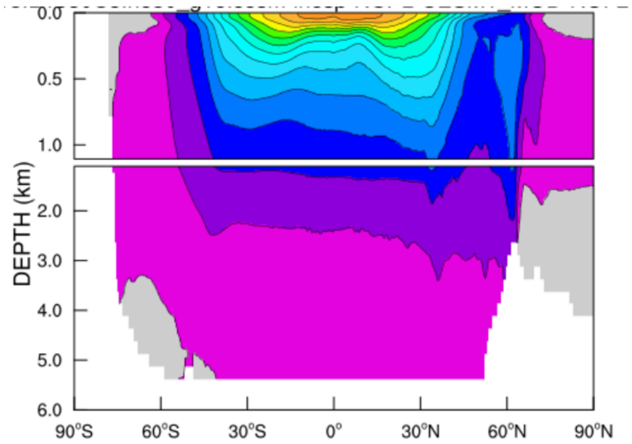
Horizontal contour plots



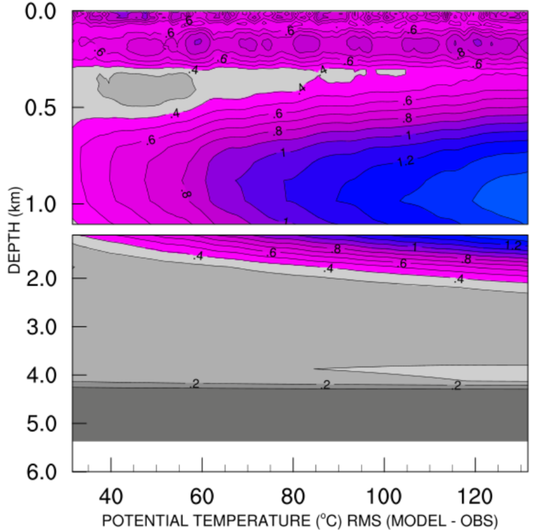
Velocity vector fields



3D fields, zonal averaged



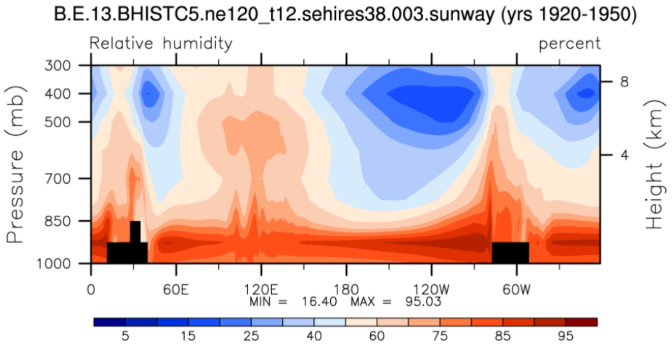
Horizontal mean vs time and depth



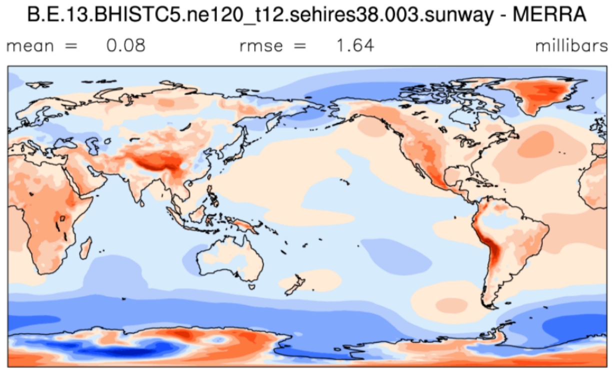
Example diagnostics page:
http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.rcp8_5.1deg.001/ocn_2080-2099-b40.20th.track1.1deg.005/popdiag.html

Atmospheric diagnostics

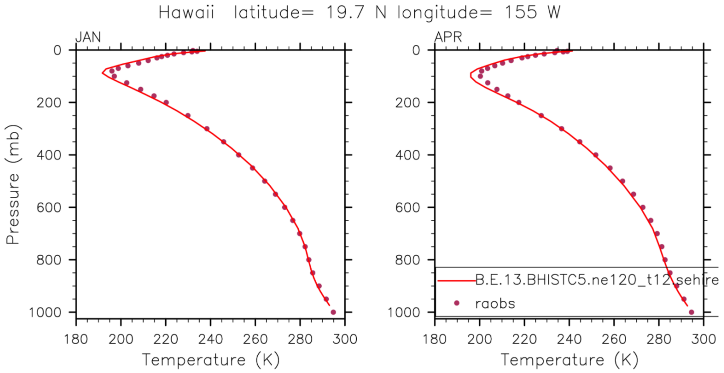
Vertical Contour Plots



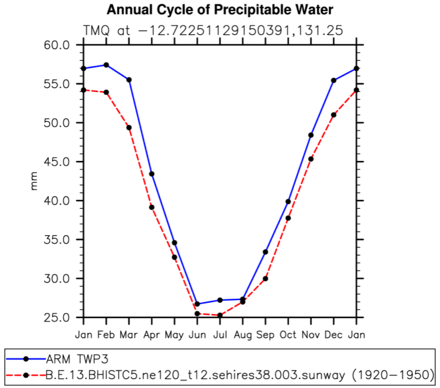
Horizontal Contour Plots



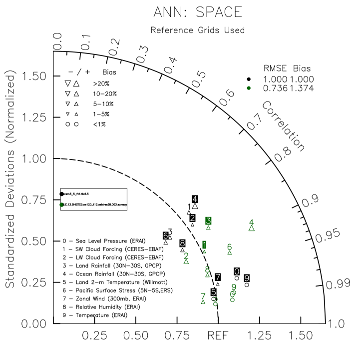
Vertical profile at a station



Annual cycle plots



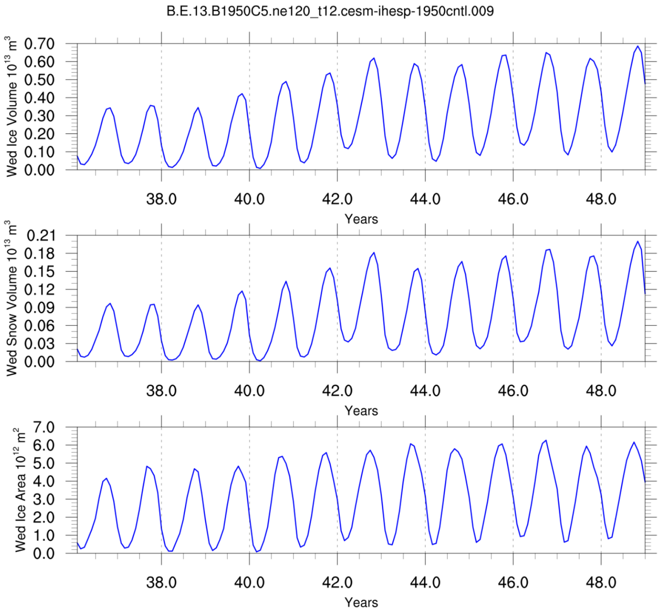
Taylor diagrams



Example diagnostics page:
http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.rcp8_5.1deg.001/atm_2010-2029-b30.099a/

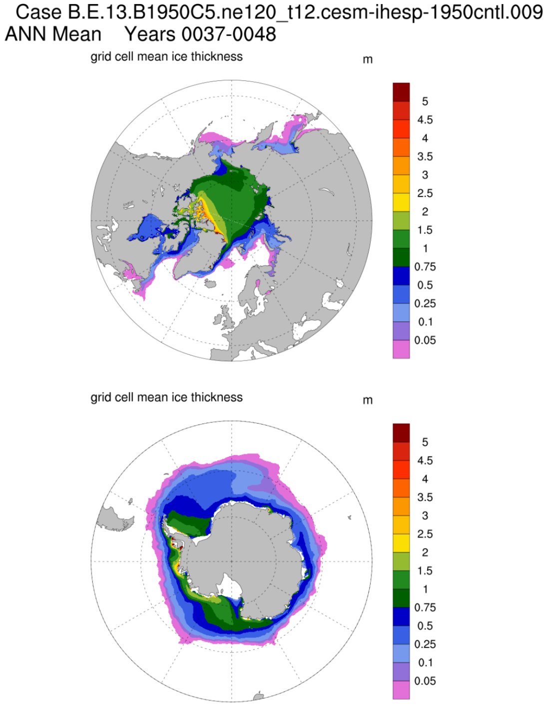
Ice diagnostics

NH & SH Timeseries Plots

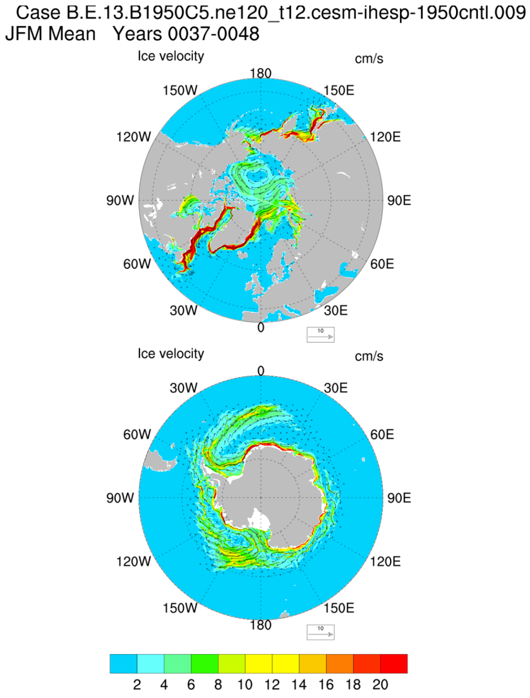


Weddell Sea monthly means

NH & SH Contour Plots



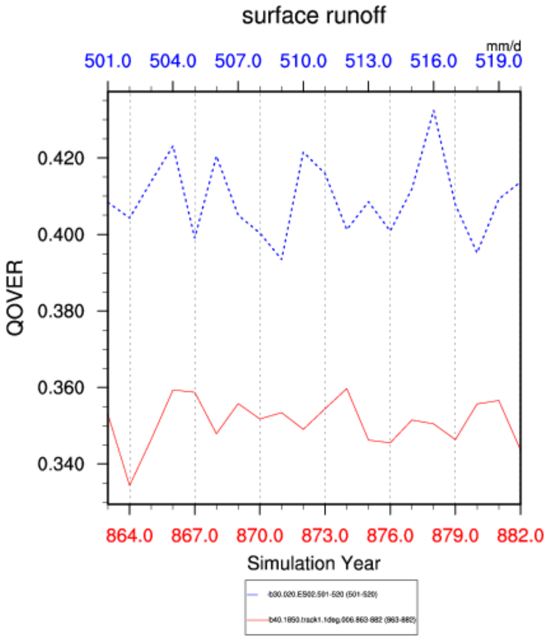
NH & SH Vector Plots



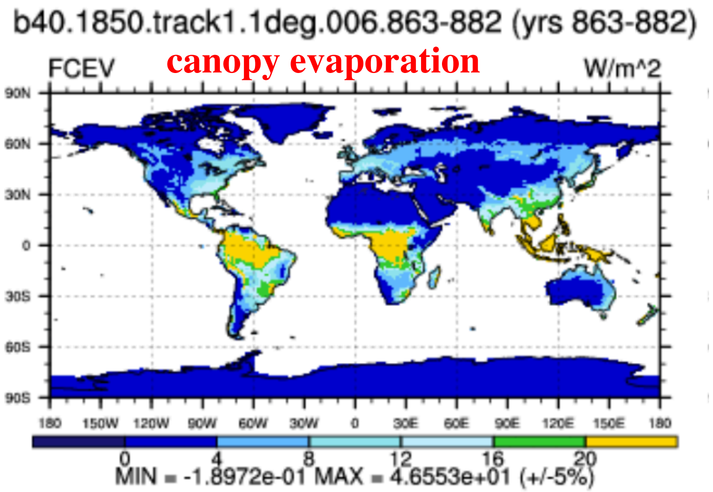
Example:
http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.1850.track1.1deg.006a/ice_1050-1079-obs/all_plots.html

Land model diagnostics

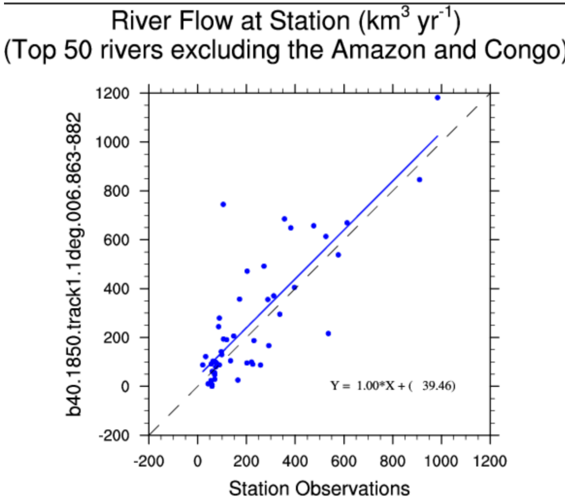
Line Plots



Horizontal Contour Plots



Scatter Plots



Example diagnostics page:
http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.1850.track1.1deg.006/lnd_863-882-b30.020.ES02/setsIndex.html

Obtaining/installing the CESM diagnostics tool

https://github.com/abishekg7/CESM_postprocessing

- Forked from NCAR’s github.
- Incorporates Ada-specific configuration created by Alper Altuntas
- Simplified installation process

Installation and Usage instructions

0. Load all necessary modules

```
module purge
ml Miniconda2/4.3.21
ml NCL/6.6.2-foss-2018b
ml NCO/4.7.9-foss-2018b
ml netCDF/4.6.1-foss-2018b-cdf5
```

1. Clone CESM_postprocessing

```
git clone https://github.com/abishekg7/CESM_postprocessing.git
cd CESM_postprocessing
export POSTPROCESS_PATH=`pwd`
```

2. Install virtual environment:

```
./create_python_env -machine ada
```

Creating a post-processing case on Ada

1. Activate the post-processing virtual environment

```
ml purge
ml Miniconda2/4.3.21
ml Python/2.7.15-foss-2018b
export POSTPROCESS_PATH=/scratch/training/CESM_post/CESM_postprocessing
source activate $POSTPROCESS_PATH/cesm-env2
```

2. Create a post-processing case directory in your scratch directory

```
cd $SCRATCH
create_postprocess -case [your-case-directory]
```

Exercise: Try the above commands on Ada

Post-processing workflow

1. Edit configuration files
   - env_postprocess.xml
   - env_diags_ocn.xml, env_diags_atm.xml, etc
2. Edit job submission scripts (ocn_averages, ocn_diagnostics, etc)
   - Change project account, num_cores, logging level, etc
3. Generate averaged files

   bsub < ocn_averages
4. Generate diagnostics

   bsub < ocn_diagnostics

Variables in env_postprocess.xml

Variable	Purpose
CASEROOT	Location of caseroot. Same as PP_CASE_PATH for standalone postprocessing
PP_CASE_PATH	Disk location of postprocessing case directory
CASE	CESM Case name
DOUT_S_ROOT	CESM model output directory. Input for post-processing
<COMP>_GRID	String specifying grid for component <COMP>
GENERATE_TIMESERIES	If True, converts history to timeseries files
GENERATE_AVGS_<COMP>	If True, calls averaging script for component <COMP>
GENERATE_DIAGS_<COMP>	If True, calls diagnostics script for component <COMP>

Grid descriptions

http://www.cesm.ucar.edu/models/cesm1.0/cesm/cesm_doc_1_0_4/a3714.html

Grid descriptor	Type of grid
1.9x2.5 or f19	Regular lon/lat finite volume grids of approximately 1.9 and 2.5 degree lon and lat respectively.
T85	Spectral lon/lat grids with 85 as the spectral truncation value for the resolution
ne30np4	Cubed sphere resolutions where X and Y are integers. The short name is generally ne[X]
pt1	Single pt grid
gx1v6 or g16	Displaced pole grid of approximately 1 degree resolution. 6 denotes the grid version.
tx0.1v2	Tripole grid of approx 0.1 degree resolution. 2 denotes the grid version.

For CESM2 -> http://www.cesm.ucar.edu/models/cesm2/config/2.1.3/grids.html

Sample values for env_postprocess.xml

Variable	Purpose
CASEROOT	/scratch/training/CESM_post/pp_cases/pp_control
PP_CASE_PATH	/scratch/training/CESM_post/pp_cases/pp_control
CASE	b.e13.B1950C5.ne30_g16.ihesp24_1950cntl.002
DOUT_S_ROOT	/scratch/training/CESM_post/datasets/b.e13.B1950C5.ne30_g16.ihesp24_1950cntl.002
ATM_GRID	ne30
OCN_GRID	gx1v6
ICE_GRID	gx1v6
LND_GRID	ne30

Variables in env_diags_ocn.xml

Variable	Purpose
OCNDIAG_DIAGOBSROOT	/scratch/training/CESM_post/obs_root
OCNDIAG_netcdf_format	netcdfLarge (64-bit)
OCNDIAG_MODEL_VS_OBS	TRUE
OCNDIAG_MODEL_VS_CONTROL	FALSE
OCNDIAG_MODEL_TIMESERIES	TRUE
OCNDIAG_YEAR0	35
OCNDIAG_YEAR1	40

Exercise – generating averages

1. Create a new postprocessing case for the transient case

2. Your env_postprocess.xml and env_diags_ocn.xml are for a control case. Make necessary changes for the transient case

3. Submit ocn_averages

4. Monitor log file at logs/ocn_averages.log.<timestamp>

Short break!
(15 minutes)

Files after averaging step (check proc/ocn/)

1. 12 Climatology files (averaged for each month)
   - **$CASE.$YEAR0-$YEAR1.<MONTH>_climo.nc**
2. Yearly-averaged files for every year
   - **$CASE.$YEAR0.nc**
3. Time average (Single file)
   - **tavg. $YEAR0-$YEAR1.nc**
4. Monthly average (Concatenation of the climatology files)
   - **mavg. $YEAR0-$YEAR1.nc**

Exercise – generating diagnostics

1. Turn on the correct diagnostic sets in env_diags_ocn.xml

2. Check the ocn_diagnostics script for correctness

3. Submit ocn_diagnostics

Summing up the CESM Postprocessing tool

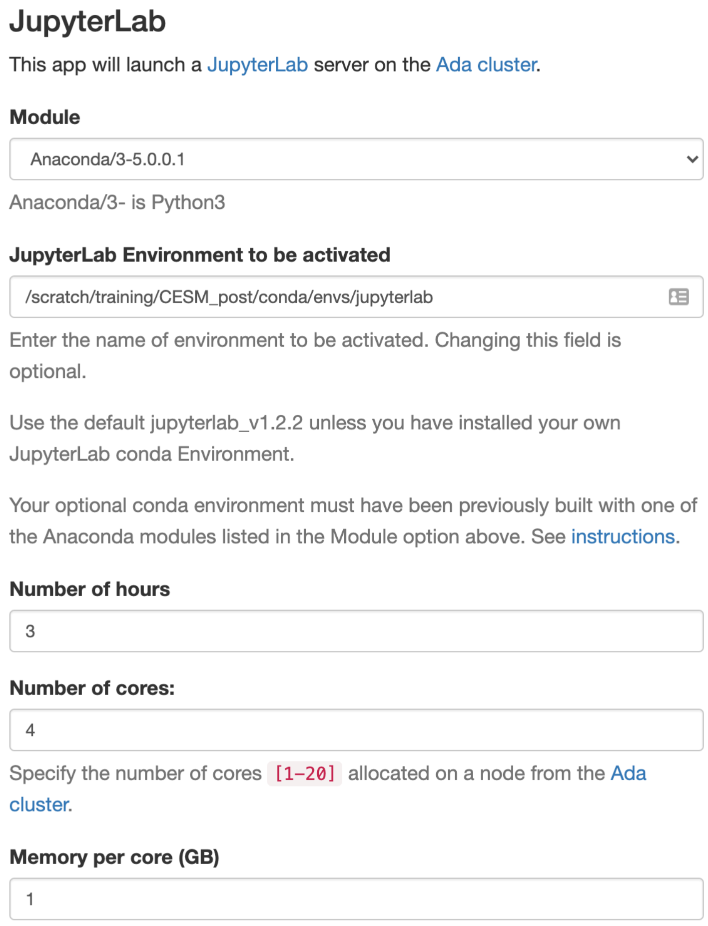
- PyAverager, PyReshaper, etc significant performance improvements over serial analysis scripts
- Works even better for high-resolution model output
- Diagnostic sets provide a birds eye view of how the model run is looking
- Can use your own Matlab/NCL scripts to visualize the generated climatology .nc files

Some limitations of the CESM PP tool

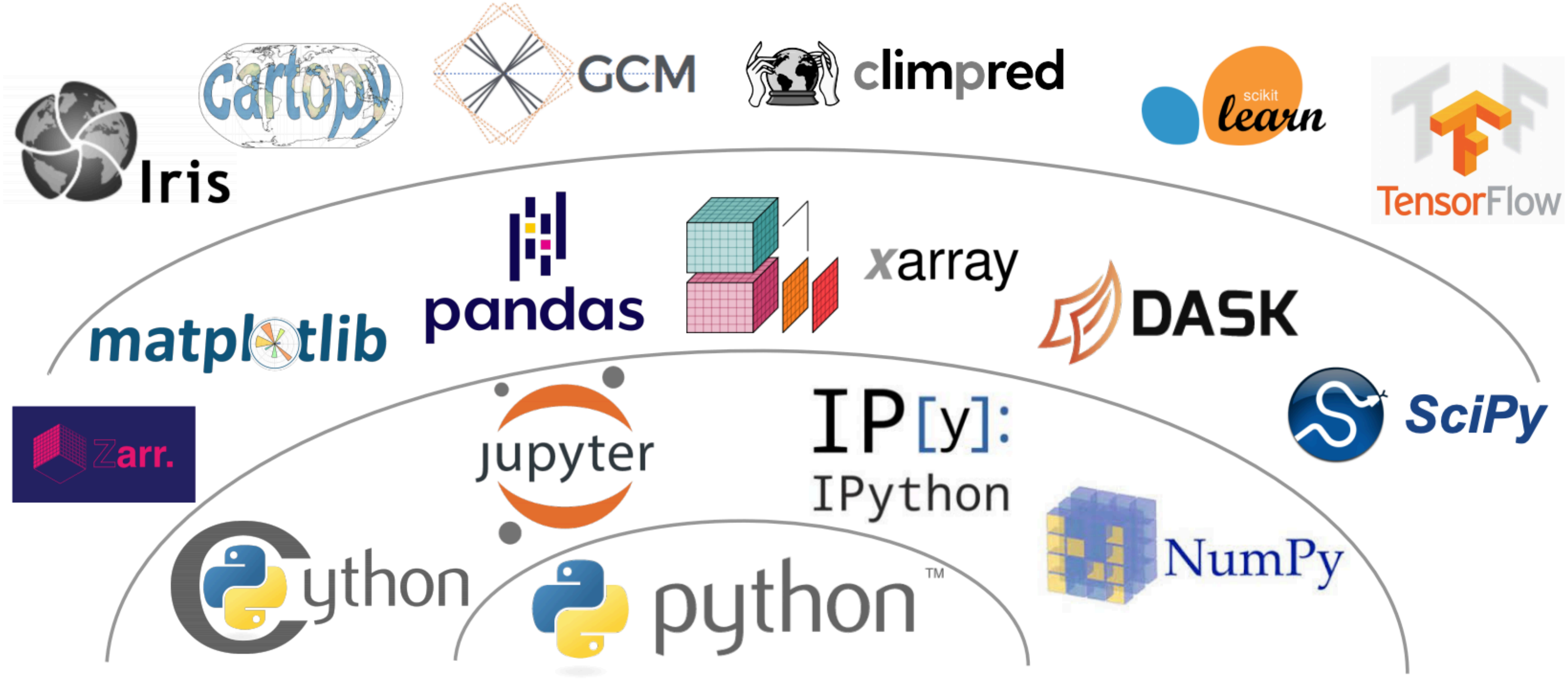
- Patchwork of many different languages and components.
- Limited help/documentation
- Adding new diagnostic sets can be a challenge
- Porting to a new machine can be tricky
- Doesn’t directly interface with CMIP6 datasets on the cloud/ESGF

Launching a JupyterLab notebook

- Go to https://portal.hprc.tamu.edu/
- Choose Ada, then Interactive Apps -> JupyterLab from the top pane
- Enter path to JupyterLab environment **/scratch/training/CESM_post/conda/envs/jupyterlab**
- Request 3 hours on 4 cores, 1 GB memory per core (4 GB in total)
- Click Launch



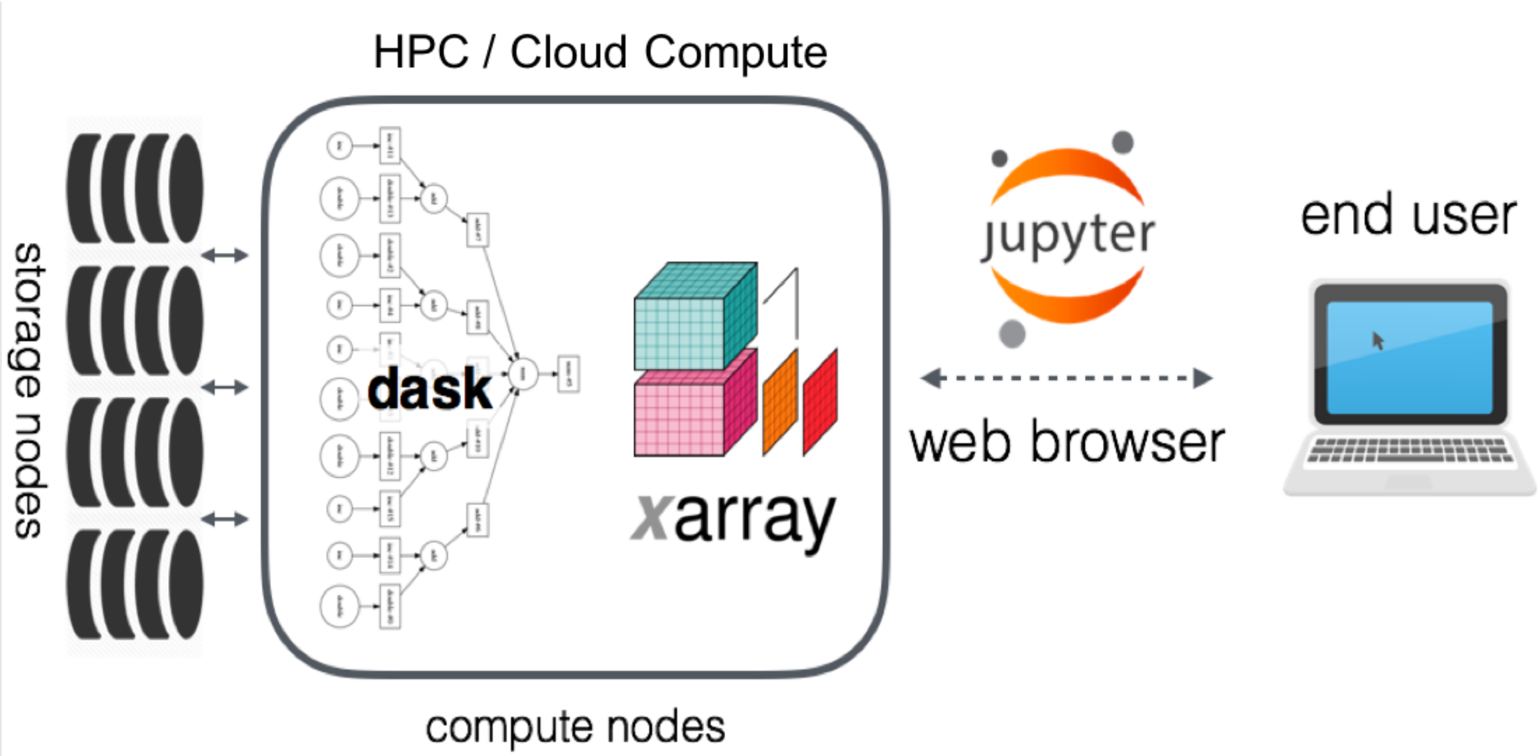
Pangeo framework



Ryan Abernathey

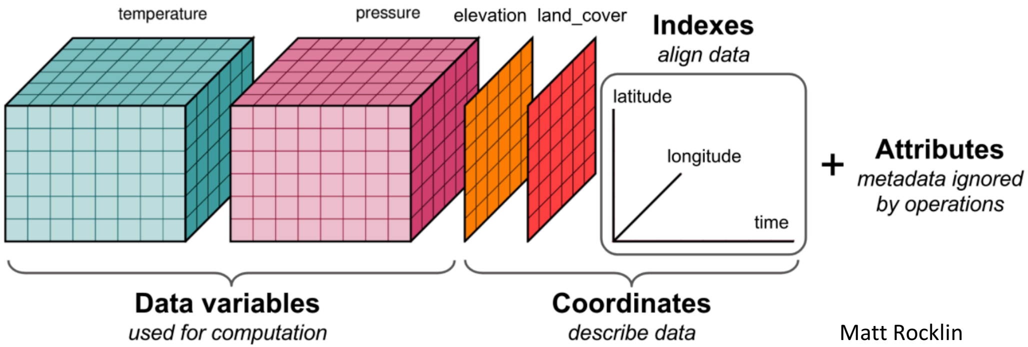
Pangeo

https://pangeo.io/architecture.html



xarray

- Applies labels like dimensions, coordinates, data variables and attributes on top of raw NumPy arrays. (See also: Pandas)
- xarray.Dataset is an in-memory representation of the netCDF file format
- xarray works seamlessly with the dask library to enable parallel computations more easily



Matt Rocklin

[2]:
```
import xarray as xr
ds = xr.open_dataset('/scratch/training/CESM_post/datasets/b.e13.B1950C5.ne30_g16.ihesp24_1
ds
```

[2]: xarray.Dataset

▸ Dimensions: (d2: 2, **lat_aux_grid**: 395, moc_comp: 3, **moc_z**: 61, nlat: 384, nlon: 320, **time**: 1, transport_comp: 5, transport_reg: 2, **z_t**: 60, **z_t_150m**: 15, **z_w**: 60, **z_w_bot**: 60, **z_w_top**: 60)

▾ Coordinates:

transport_comp...	(transport_comp)	|S256	...
transport_regions	(transport_reg)	|S256	...
time	(time)	object	0038-05-01 00:00:00
z_t	(z_t)	float32	500.0 1500.0 ... 512502.8 537500.0
z_t_150m	(z_t_150m)	float32	500.0 1500.0 ... 13500.0 14500.0
z_w	(z_w)	float32	0.0 1000.0 ... 500004.7 525000.94
z_w_top	(z_w_top)	float32	0.0 1000.0 ... 500004.7 525000.94
z_w_bot	(z_w_bot)	float32	1000.0 2000.0 ... 549999.06
lat_aux_grid	(lat_aux_grid)	float32	-79.48815 -78.952896 ... 90.0
moc_z	(moc_z)	float32	0.0 1000.0 ... 525000.94 549999.06
ULONG	(nlat, nlon)	float64	...
ULAT	(nlat, nlon)	float64	...
TLONG	(nlat, nlon)	float64	...
TLAT	(nlat, nlon)	float64	...

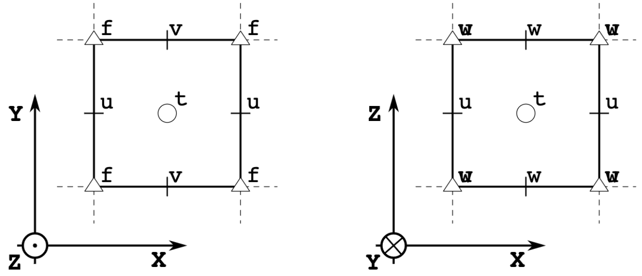
▸ Data variables: (113)

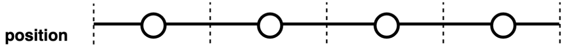
▸ Attributes: (11)

GCM

High Performance Research Computing
DIVISION OF RESEARCH

- xarray doesn't implicitly understand GCM grids
- xgcm wraps xarray to add an understanding of grid topology
- Implements spatial derivative operators
- Understands only C-grids for now, but other works are in progress





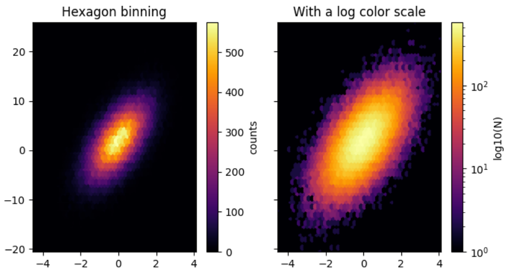
position									
center		f[0]		f[1]		…		f[n-1]	
left	f[0]		f[1]		…		f[n-1]		
right			f[0]		f[1]		…		f[n-1]
inner			f[0]		…		f[n-2]		
outer	f[0]		f[1]		…		f[n-1]		f[n]

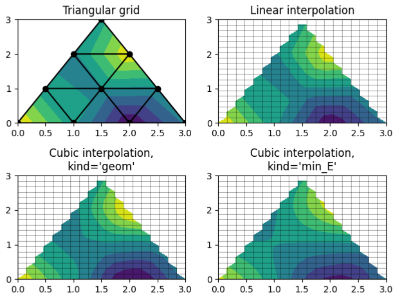
The different possible positions of a variable `f` along an axis.

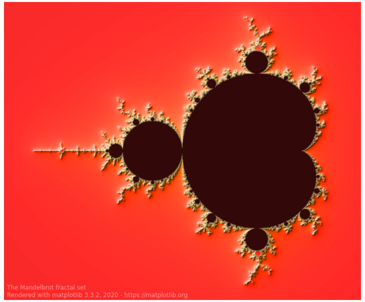
https://xgcm.readthedocs.io/en/latest/grids.html

matplotlib

A comprehensive library for creating static, animated, and interactive visualizations in Python.



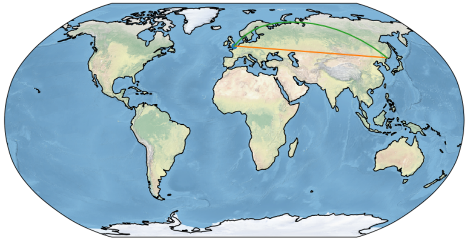


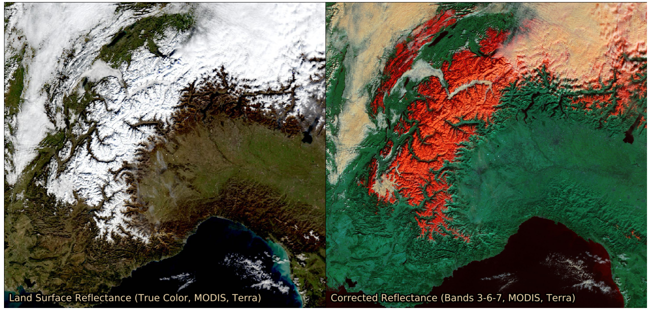


https://matplotlib.org/gallery/

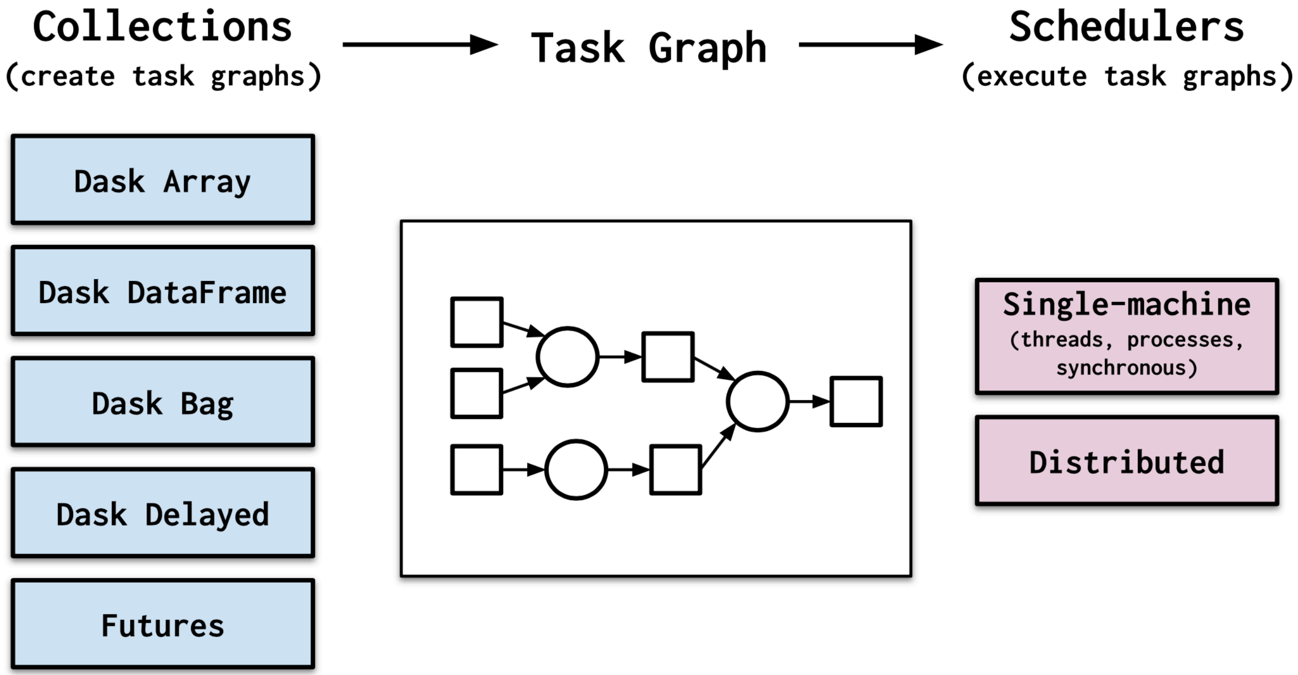
cartopy

Cartopy adds understanding of map projections to matplotlib plots



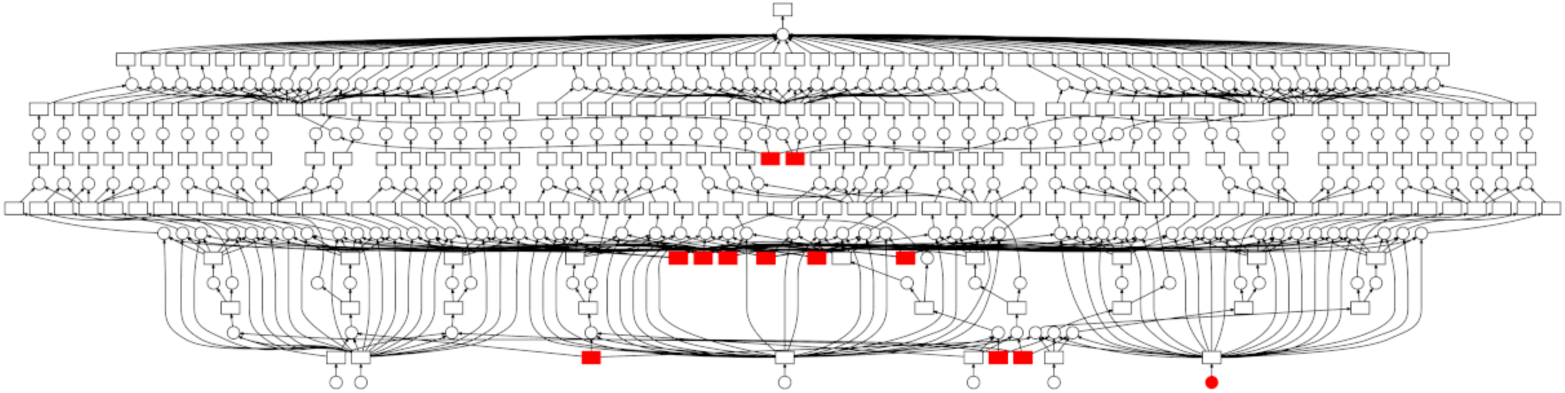


https://scitools.org.uk/cartopy/docs/latest/gallery/index.html



https://docs.dask.org/en/latest/

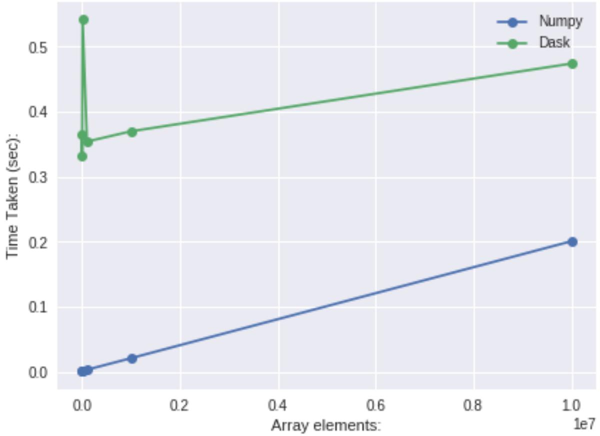
Dask task graph

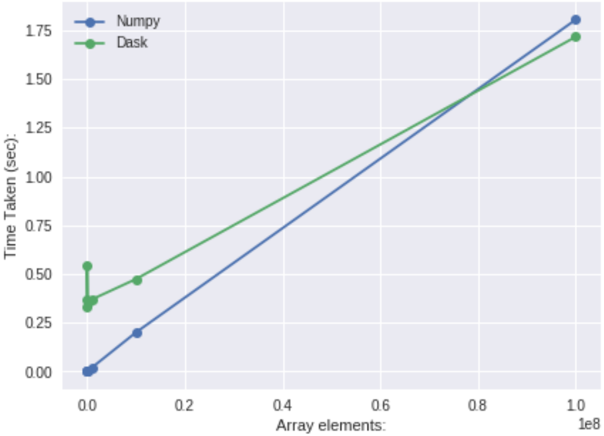


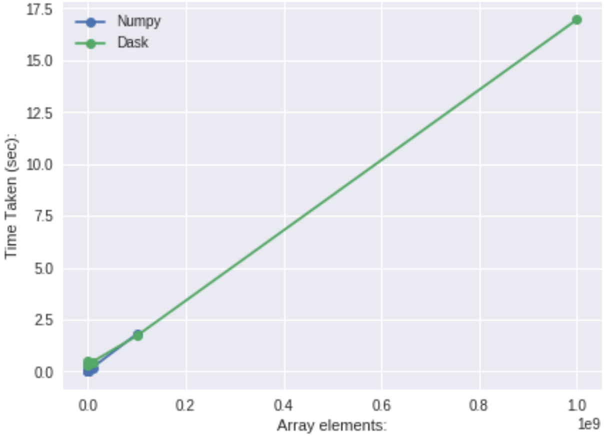
https://towardsdatascience.com/speeding-up-your-algorithms-part-4-dask-7c6ed79994ef

Dask vs NumPy

- NumPy is faster than Dask for a smaller problem size
- For larger problems, Dask achieves better scalability
- Larger datasets require correspondingly large amounts of memory with NumPy, and this is where dask's lazy loading shines







https://towardsdatascience.com/speeding-up-your-algorithms-part-4-dask-7c6ed79994ef

Download tutorial notebooks

- ssh <username>@ada.tamu.edu

- cd $SCRATCH

- Clone the notebook repository from github

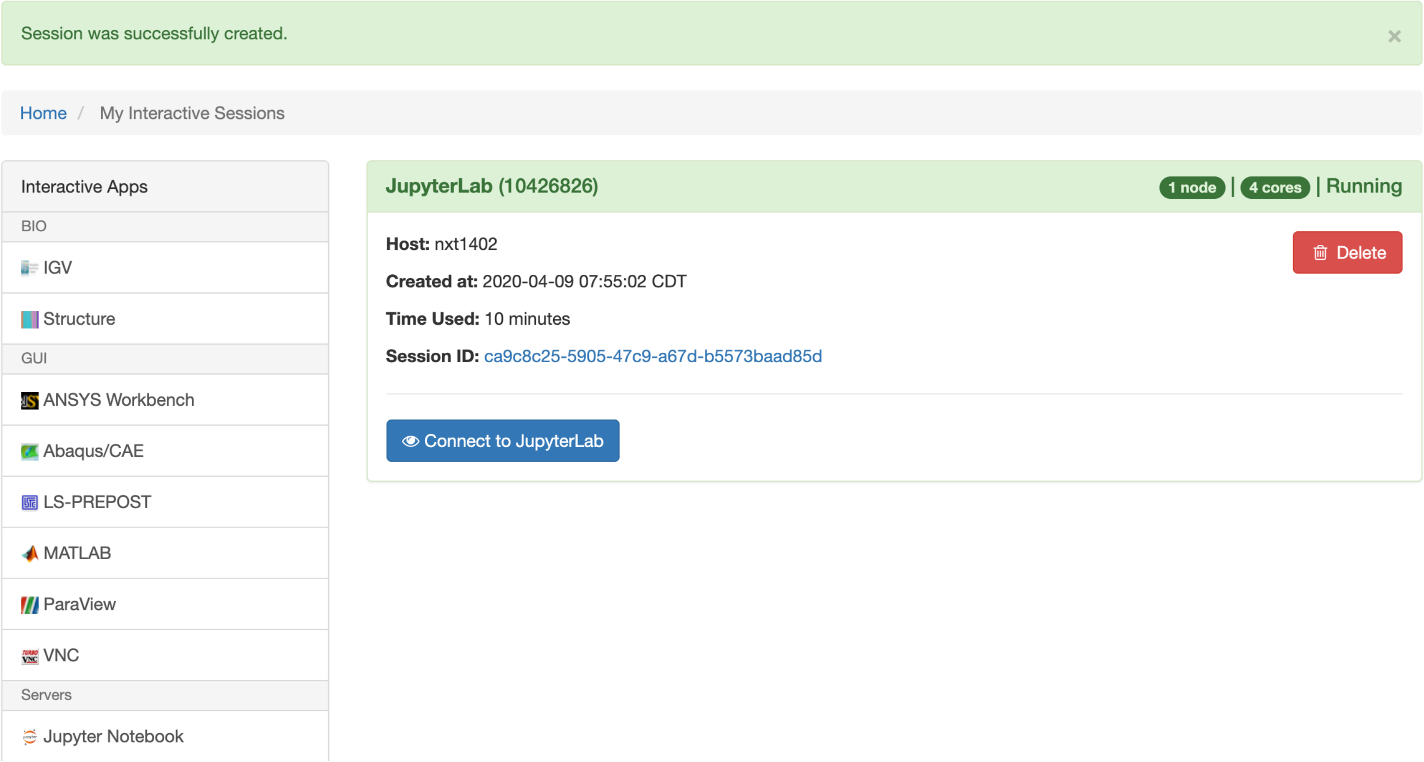
  git clone https://github.com/abishekg7/pangeo_binders.git

  OR

  Copy notebooks from Ada

  cp -r /scratch/training/CESM_post/pangeo_binders .

Launch JupyterLab notebook



CESM Postprocessing vs Pangeo – Key Takeaways

High Performance Research Computing
DIVISION OF RESEARCH

- The CESM Postprocessing tool offers a more reliable and comprehensive set of parallelized post-processing functions
  - Can use PyAverager and PyReshaper independently

- The Pangeo framework rethinks how we analyze large datasets
  - Dask enforced lazy-loading + parallelization
  - In its developmental stages, and will take a few more years to reach the breadth of NCL
  - For newer analysis tools development, consider using Pangeo

Additional resources

- CESM diagnostics tool tutorials
  - http://www.cesm.ucar.edu/events/tutorials/2019/files/Practical3-phillips.pdf
  - http://www.acacia.ucar.edu/events/2019/ctsm/files/practical21-oleson.pdf
  - http://www.cesm.ucar.edu/events/workshops/ws.2016/presentations/sewg/bertini.pdf
- GeoCAT
  - https://geocat.ucar.edu/pages/software.html
- xroms (by Rob Hetland, Kristen Thyng, TAMU Oceanography)
  - https://github.com/hetland/xroms
- Pangeo forums
  - http://discourse.pangeo.io/

Acknowledgements

- Lisa Perez and the HPRC team
- Kristen Thyng, TAMU Oceanography
- Sanjiv Ramachandran and Dapeng Li, iHESP
- Fred Castruccio and Alper Altuntas, NCAR

Questions?

Some ocean diagnostics plots

Type of plot	Quantity
2D Surface flux fields	SENH_F, LWUP_F, EVAP_F, PREC_F
2D Surface fields	SSH, SST, etc
3D fields (Zonally-averaged)	TEMP, SALT, IAGE
Fields at various depth levels	TEMP, SALT, UVEL, VVEL, WVEL
Vector fields at various depth levels	velocity
MOC	

Dask client-scheduler-worker